Learning Multiple Intent Representations for Search Queries

Helia Hashemi, Hamed Zamani, and W. Bruce Croft
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
Amherst, MA, United States
{hhashemi,zamani,croft}@cs.umass.edu

ABSTRACT

Representation learning has always played an important role in information retrieval (IR) systems. Most retrieval models, including recent neural approaches, use representations to calculate similarities between queries and documents to find relevant information from a corpus. Recent models use large-scale pre-trained language models for query representation. The typical use of these models, however, has a major limitation in that they generate only a single representation for a query, which may have multiple intents or facets. The focus of this paper is to address this limitation by considering neural models that support multiple intent representations for each query. Specifically, we propose the NMIR (Neural Multiple Intent Representations) model that can generate semantically different query intents and their appropriate representations. We evaluate our model on query facet generation using a large-scale dataset of real user queries sampled from the Bing search logs. We also provide an extrinsic evaluation of the proposed model using a clarifying question selection task. The results show that NMIR significantly outperforms competitive baselines.

ACM Reference Format:
Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2021. Learning Multiple Intent Representations for Search Queries. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3459637.3482445

1 INTRODUCTION

Neural network approaches have shown promising results in many information retrieval (IR) tasks [21], including but not limited to ad hoc retrieval [20], web search [47], personal search [75], and conversational search [23, 55, 74]. An emerging recipe for achieving state-of-the-art effectiveness in neural IR models involves utilizing large pre-trained language models (LLMs), e.g., BERT [14] and BART [39], for representing user inquiries and documents [43]. Although these representations benefit from well-designed attention mechanisms and have led to significant performance improvements in many IR and NLP tasks, they have their own shortcomings in deployment for some certain tasks. For instance, in query representation learning, which is a core IR problem, the current common practice is to use the query text as the LLM's input and produce a single representation for the query, e.g., see [28, 72]. However, as is widely accepted [60], each query may be associated with multiple intents.[1] Intuitively, learning a single representation for each query is equivalent to learning a centroid representation of all potential query intents. We argue that the centroid representation is not necessarily representative of either individual query intents or the query itself as a whole. This is because the single representation strategy causes information loss for individual query intents and cannot be semantically inclusive for all query intents. Consequently, it cannot be optimal for many IR applications, including query facet generation, query disambiguation, search result diversification, and clarification in web and conversational search engines.

In this paper, we address this issue by proposing a *general framework* for learning multiple far-flung and widely distributed representations for a query such that each representation addresses one of its potential intents. Our framework, called NMIR, is designed based on a neural encoder-decoder architecture, and is optimized such that the generic query representations produced by the encoder are transformed to multiple remotely distributed representations, each associated with a query intent. We study both parametric and non-parametric variations of the framework. In the former, the model assumes that the number of representations per query is given, while the latter dynamically identifies the number of representations for each query.

We optimize our framework based on the following hypothesis: if the query encoder can accurately learn multiple query intent representations, therefore the decoder should be able to accurately generate all intent descriptions. On this basis, the training objective in NMIR is to maximize the likelihood of generating the query intent descriptions (or facets). In order to improve the efficiency of our framework, we introduce an *asynchronous* training strategy in which one process is responsible for model training and another one adjusts the enforcement conditions that obligates the model to generate widely distributed representations.

NMIR has applications in a wide range of IR tasks reviewed in Section 4. We perform extensive experiments for extrinsic evaluation of the model in two real-world downstream tasks: query facet generation and search clarification. We demonstrate significant improvements compared to competitive baselines using offline evaluation on reusable test collections in addition to manual pairwise comparison with the baseline using three trained annotators.

2 RELATED WORK

Learning accurate query representations is a core problem in neural information retrieval. It has applications to query-level tasks, such as query classification [42, 77], query re-writing [24], query auto-completion [3, 51, 69], and query suggestion [12]. It is also an important component in late combination neural ranking models [13, 21], such as DSSM [25], SNRM [79], ColBERT [28], and ANCE [72]. Existing neural ranking models learn a single representation for a given

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8446-9/21/11...$15.00
https://doi.org/10.1145/3459637.3482445

[1] In this paper, query intent and facet are used interchangeably.

search query. However, search queries often carry multiple intents. Therefore, such models theoretically summarize all the query intent representations by their centroid representation. We believe that neural models should go beyond a single query representation in order to effectively address various information retrieval tasks. In more detail, this paper proposes a new task of learning far-flung representations for a query input in order to model its various intents. In this section, we review prior research on related topics including query representation learning, query facet generation, and query reformulation in addition to search result diversification and search clarification.

2.1 Query Representation.

Query representation is at the core of IR models. For instance, in vector space models based on term matching [59], queries are represented based on term occurrences and frequencies in the queries and general statistics of the collection. Several models have focused on improving this representation, mainly with a focus on addressing the vocabulary mismatch problem [18], for example through query expansion and (pseudo-) relevance feedback [8, 37, 58, 83].

In machine learning based approaches, query representations are often learned. Latent semantic indexing (LSI) [11] is an early unsupervised method for query and document representation learning that uses singular value decomposition over a matrix of term frequencies in the given texts to embed them into a latent space. The same basic concepts are also used in many neural text representation learning models. For instance, word2vec [46] and GloVe [52] learn unsupervised word representations by predicting words given their adjacent words or vice versa in a large text collection. Early attempts to use word embedding models for information retrieval mainly focused on query expansion [34, 76] and document expansion or language model smoothing [19].

Zamani and Croft [77] proposed the first model for deriving query representations from the learned embedding vectors of individual query terms. They introduced a theoretical framework for query representation and showed that a maximum likelihood optimization approach for query representation would lead to averaging the embedding vectors of query terms, if no more information is available. In their follow up work [78], the authors suggested to learn IR-specific word and query embeddings by predicting the words appearing in (pseudo-) relevant documents in response to each query. Diaz et al. [15] alternatively suggested to train word2vec models on local context, i.e., the top retrieved documents in response to the query. Later on, Zhang et al. [85] showed that the relevance-based word embedding of Zamani and Croft [78] can be further trained on clicked documents obtained from a search engine's log, and proposed a generic query representation model that is trained using various implicit feedback signals, e.g., clicks, with multi-task learning. More recently, large-scale contextual embedding models, such as BERT [14], are used to represent queries and documents for a range of IR tasks [43]. These models require further fine-tuning using supervised signals for the downstream task to perform effectively.

All the query representation learning methods pointed out in this section produce a single representation for each query. This single representation can be a single vector and/or a single vector per query term. Therefore, they cannot be used for representing and generating different query intents. This paper, on the other hand, introduces a model that learns multiple representations per query. This would lead to multiple applications that cannot be solved using the existing techniques (see Section 4).

2.2 Query Facet Extraction and Generation.

Early work on facet extraction and/or generation [9, 29, 35, 40, 64] focused on applications like e-commerce and digital libraries, where facets can be extracted from existing metadata or taxonomies. These approaches are not practically extendable to large-scale open-domain settings.

Besides leveraging taxonomies and external resources, some models extract facets by global analysis of the entire search corpus [9, 64]. However, the heterogeneous nature of many search collections, such as the web content, makes such approaches not adoptable [66]. To address this issue, approaches based on local analysis were invented [16, 30, 31]. They extract query facets from the top retrieved documents in the search result list for the query. Notably, Kong and Allan [30, 31, 32] developed a graphical model based approach for facet extraction. They showed that the optimization of their model is an NP-hard problem and thus proposed two approximations (called QFI and QFJ) based on different simplifying assumptions on computing the joint probabilities in the proposed graphical model. Later on, Dou et al. [17] introduced QDMiner that extracts facets with a hybrid approach.

Although query facet generation models do not explicitly learn query representations, they are in root related to representing different query intents. Therefore, we used query facet generation in one of our experiments to evaluate our model. We compare against the state-of-the-art QFI, QFJ, and QDMiner variations [17, 32] and demonstrate the effectiveness of the proposed solution.

2.3 Search Result Diversification, Query Reformulation, and Clarification.

Search queries do not always clearly express the users' information needs. IR scientists categorized these types of queries into two types; ambiguous and underspecified queries [10]. Ambiguous queries have more than one interpretation, while, underspecified queries have one interpretation but with several sub-topics. Search result diversification and intent clarification are two major approaches for addressing ambiguous and underspecified queries.

Search result diversification re-ranks the result list in order to cover as many query intents as possible. Hence, many query intents can be addressed by a single result list. To do so, most of the existing methods perform an initial retrieval and then select some documents from top k retrieved set based on some criteria [62]. These methods can be categorized in two groups of "implicit" and "explicit" approaches. Implicit approaches choose documents different from those which have been chosen previously without the explicit modeling of facets [4, 84]. Maximal marginal relevance (MMR) [4] is a simple yet effective greedy algorithm for implicit diversification. On the other hand, explicit approaches attempt to model query sub-topics, which are closer to our work. For example, Agrawal et al. [1] used taxonomy, and several other researchers [56, 61, 73] used query *reformulations* to model query sub-topics. Alternatively, Dang et al. [10] generated query reformulations by using anchor texts, and Carterette and Chandar [5] focused on the retrieved documents for explicit diversification by adopting relevance and topic models.

As an alternative to diversification, search engines can clarify the users' information needs by asking clarifying questions. This has applications in both web search with the traditional "ten blue link" interface [80, 82] and conversational search with limited bandwidth interfaces where search result diversification is impractical [2].

Learning multiple query intent representations has applications in both search result diversification and intent clarification. In our experiments, we also extrinsically evaluate our model using a clarifying question selection task.

3 METHODOLOGY

Training query representation learning models that are able to produce multiple widely distributed representations for each search query has not yet been explored. This is a challenging task, especially when the number of representations varies across queries. In this section, we propose a general framework for this task with an optimization solution that has roots in cluster-based IR models studied for decades [26, 33, 38, 44, 68]. Unlike prior work, NMIR takes advantage of clustering during asynchronous training in order to learn far-flung and widely distributed representations. The NMIR framework can be further employed in a wide range of downstream IR applications. Some of them are reviewed in Section 4. Task-based fine-tuning can be adopted for each downstream task.

3.1 Task Description and Problem Formulation

The task is to learn multiple widely distributed representations for each search query. We use the top retrieved documents in a search result list in response to the query as a source of evidence to find various intents of the query for representation learning. For training the model, we assume that a textual description of each query intent is available. In Section 3.3, we discuss potential solutions on obtaining such descriptions.

Before formalizing the task, we introduce our notation. Let $Q = \{q_1, q_2, \ldots, q_n\}$ be a training query set with n queries, and $D_i = \{d_{i1}, d_{i2}, \ldots, d_{im}\}$ be the top m retrieved documents in response to the query q_i using a retrieval model M. Moreover, let $F_i = \{f_{i1}, f_{i2}, \ldots, f_{ik_i}\}$ denote the set of all textual intent descriptions associated with the query q_i. k_i is the number of query intents and can vary across queries. The task is to learn k_i representations $R_i = \{R_{i1}, R_{i2}, \ldots, R_{ik_i}\}$ for the query q_i, where R_{ij} is the j^{th} representation learned for the query.

3.2 NMIR Framework: A High-Level Overview

One straightforward solution for the task is using an encoder-decoder architecture that leverages the query q_i (and the top retrieved documents) as the input and generates multiple query intent descriptions of the query by taking the top k_i most likely predictions, e.g., using beam search. However, previous work in a number of NLP tasks [67, 70] showed that these generations are often synonyms or refer to the same concept, which is in contrast to the goal of our task: learning *widely distributed* representations, each associated with a query intent. This solution generates different but semantically similar outputs, which are only related to one query intent. Hence, this approach would not serve the purpose.

Another straightforward solution is to look at the task as a sequence-to-sequence problem, similar to machine translation, and generate all the query intent descriptions concatenated with each other (and separated using a special token). The concern regarding this approach is that different intent representations are not distinguishable in the last layer of the model. In addition, most existing effective text encoding models are not able to represent long sequences of tokens, such as a concatenation of the top m retrieved documents.

The NMIR framework addresses these issues. Let $\phi(\cdot)$ and $\psi(\cdot)$ denote a text encoder and decoder pair, respectively. For every query q_i in the training set, NMIR assumes that the top retrieved documents D_i are relevant to the query and they may be relevant to different query intents. NMIR assigns each learned document representation to one of the query intent descriptions $f_{ij} \in F_i$ using a document-intent matching algorithm γ:

$$C_i^* = \gamma\big(\phi(d_{i1}), \phi(d_{i2}), \ldots, \phi(d_{im}), \phi(f_{i1}), \phi(f_{i2}), \ldots, \phi(f_{ik_i})\big)$$

where $C_i^* = \{C_{i1}^*, C_{i2}^*, \ldots, C_{ik_i}^*\}$ is a set of document sets. Each C_{ij}^* is a set of documents from D_i that are assigned to f_{ij} by γ.

NMIR then transforms the encoded general query representation to its intent representations through a query intent encoder ζ. In more detail, the representation for the j^{th} query intent is obtained using $\zeta(q_i, C_{ij}^*; \phi)$. The implementation details of components ϕ, ψ, γ, and ζ are presented in Section 3.3.

NMIR's training for a mini-batch b is based on a gradient descent-based minimization of $\mathcal{L}(b) = \frac{1}{|b|} \sum_{q_i \in b} L(q_i)$, where $L(q_i)$ is defined as follows:

$$L(q_i) = \frac{1}{k_i} \sum_{j=1}^{k_i} L_{\text{CE}}(f_{ij}, \psi(\zeta(q_{ij}^*, C_{ij}^*; \phi)))$$

where $q_{ij}^* =$ “$q_i f_{i1} f_{i2} \ldots f_{ij-1}$ <mask>...<mask>” is a concatenation of the query string, the first $j-1$ intent descriptions, and $k_i - j$ mask tokens. There is a special separation token between each of these strings. Therefore, $L(q_i)$ basically calculates the loss for generating each textual intent description, given the associated cluster C_{ij}^* and the encoded query text plus the past $j-1$ intent descriptions. This helps the model avoid generating the previous intent representations and learn widely distributed representations.

In the above loss function, L_{CE} is the cross-entropy loss borrowed from the sequence-to-sequence model [65]:

$$-\sum_{t=1}^{|f_{ij}|} \log p\Big(f_{ijt} | \psi(\zeta(q_{ij}^*, C_{ij}^*; \phi)), f_{ij1}, f_{ij2}, \ldots, f_{ijt-1}\Big)$$

where f_{ijt} is the t^{th} token in the given intent description f_{ij}.

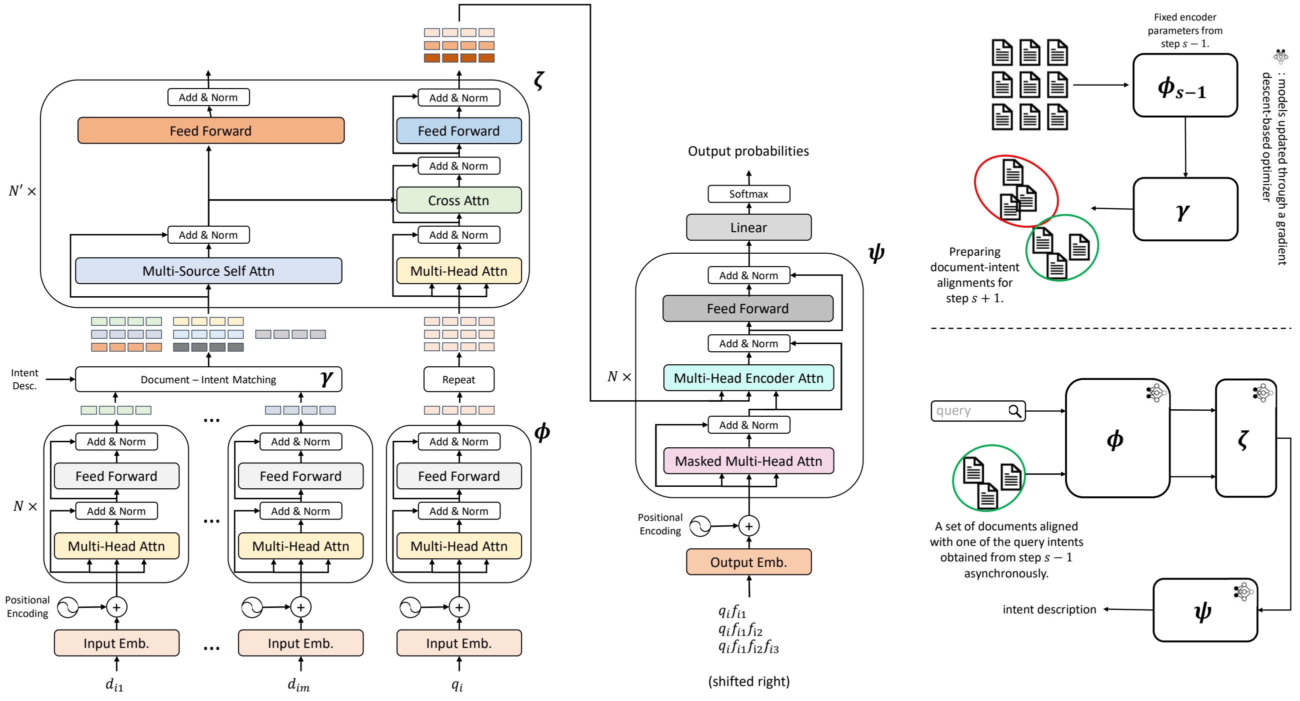
Inference. Using NMIR at inference time is partly different from the way it is used during training. To be precise, the q_{ij}^*s are constructed differently. At training, they are constructed by concatenating the query and the previous intent descriptions in order to generate the next one. While at inference, we do not have access to the intent descriptions, therefore we should construct q_{ij}^*s based on the model's output. Therefore, for the query q_i, we first feed “q_i <mask>...<mask>” to the model (the number of mask tokens is equal to $|C_i^*|$) and apply *beam search* to the decoder's output to obtain the first intent description f_{i1}'. We then use the model's output to iteratively create the input for the next step “$q_i f_{i1}'$ <mask>...<mask>” and repeat this process for $|C_i^*|$ times. As mentioned earlier, similar to the model training, the reason for including previous outputs is to avoid generating repetitive intent descriptions.

3.3 Model Implementation and Training

This subsection describes the detailed implementation of our framework for each of its components. We implemented our model using the PyTorch Lightning platform.[2]

The encoding and decoding components ϕ and ψ. As depicted in Figure 1a, we use Transformer encoder and decoder architectures for implementing ϕ and ψ, respectively. We initialize their parameters with the pre-trained BART model [39]. BART is a denoising

[2] https://www.pytorchlightning.ai/



(a) The network architecture of NMIR.

(b) Asynchronous training of NMIR.

Figure 1: (a) The network architecture of NMIR. Same background colors indicate parameter sharing. White background means that the component does not have learnable parameters. The encoder and decoder parameters (ϕ and ψ) are initialized by BART pre-trained parameters [39] consisting of N Transformer layers and are fine-tuned. (b) The asynchronous training of the NMIR framework. These two steps (above and below the dashed line) are executed on two different GPUs, and the model parameters are only updated in one of the steps, using a gradient descent-based optimizer. ϕ_{s-1} represents the encoder whose parameters are fixed and obtained from a model snapshot at step $s-1$.

autoencoder for pretraining sequence-to-sequence models. It uses standard Transformer-based encoder-decoder architecture and has been pre-trained based on adding noise to the input text and reconstructing it. In extreme cases, where the input text is corrupted to the extent that there is no information left from the original format, BART is equal to language models. We use the BART's implementation delivered by the HuggingFace's Transformer library [71].[3] In NMIR, the decoder's cross attention is the output of the intent encoder ζ for each query intent (see Figure 1a).

The intent encoding component ζ. As shown in Figure 1a, the intent encoding component $\zeta(q^*_{ij}, C^*_{ij}; \phi)$ is implemented using N' layers of the recently developed Guided Transformer model of Hashemi et al. [23]. Guided Transformer is used for influencing an input representation by the *guidance* of some external information. In our case, we use $\phi(q^*_{ij})$ as the input representation and $\phi(d): \forall d \in C^*_{ij}$ as the external information. In fact, Guided Transformer uses self attention on the input tokens (the query), self attention on each external resource (each document in C^*_{ij}), and a cross attention from the document representations to the query representation. This cross attention mechanism enables the model transform the generic query representation to a query intent representation.

The document-intent matching component γ. Inspired by work on multi-sense word embedding [41, 48], for document-intent matching based on the encoded representations, we develop an algorithm that clusters the learned representations and assigns each cluster to an intent description. In more details, NMIR encodes all the top retrieved documents and creates k_i clusters, using a clustering algorithm. Therefore, we have:

$$C_i, \mathcal{M}_i = \text{cluster}(\phi(d_{i1}), \phi(d_{i2}), \ldots, \phi(d_{im}))$$

where $C_i = \{C_{i1}, C_{i2}, \ldots, C_{ik_i}\}$ denotes a set of clusters and each C_{ij} contains all the documents in the j^{th} cluster associated with the query q_i. $M_i = \{\mu_{i1}, \mu_{i2}, \ldots, \mu_{ik_i}\}$ is a set of all cluster centroids such that $\mu_{ij} = \text{centroid}(C_{ij})$. In our implementation, we use K-Means [22] for clustering in this step, due to its simplicity and efficiency. K-Means has been successfully used in a number of IR applications [26, 33, 44, 68]. Note that K-Means requires the number of clusters as input. The number of clusters for q_i at the training time is given by the number of intent descriptions (i.e., k_i). However, this value is unknown at inference time. In our experiments, we consider two cases. In the first case, we assume that the number of clusters at test time is equal to a tuned hyper-parameter k^* for all queries. In the second case, we replace the K-Means algorithm by a non-parametric version of K-Means [45]. This algorithm basically starts with creating one cluster based on a minimum document similarity threshold. Once

[3] https://huggingface.co/transformers/

the first cluster is created, the same process would be repeated for the rest of documents that are not yet assigned to any clusters. For more information on non-parametric K-Means, we refer the reader to [45].

The component γ requires a one-to-one assignment between the cluster centroids and the query intents in the training data. The assignment needs to be one-to-one, since otherwise all clusters may be assigned to a single most dominant query intent, and thus the model would not learn to generate far-flung query representations. Therefore, NMIR uses the following injective surjective function, called the intent identification function $\mathcal{I}$:

$$\mathcal{I}(M_i,F_i) = \arg\max_{M' \in \text{perm}(M_i)} \sum_{j=1}^{k_i} \text{sim}(\phi(f_{ij}),\mu'_j)$$

where perm$(\cdot)$ returns all permutations of a given set and each $M' = [\mu'_1,\mu'_2,...,\mu'_{k_i}]$ denotes a permutation of cluster centroids in M_i. The function sim$(\cdot,\cdot)$ denotes a similarity function. We use inner product to compute the similarity between an intent representation and a cluster centroid. Therefore, let $M^*_i = [\mu^*_{i1},\mu^*_{i2},...,\mu^*_{ik_i}]$ be the output of $\mathcal{I}(M_i,F_i)$ and $C^*_i = \{C^*_{i1},C^*_{i2},...,C^*_{ik_i}\}$ be their associated clusters. The component γ returns C^*_i.

Note that the γ is not differentiable and cannot be part of the network for gradient descent-based optimization. Our asynchronous training (presented below) addresses this issue by taking γ out of the optimization process and moving it to an asynchronous process (see Figure 1b). Another important point is that there is no need to call the function $\mathcal{I}$ at inference time, because the order of the clusters does not matter, while it matters for training as it helps us compute the loss function.

Asynchronous training. As widely known, the training speed of deep learning models can be greatly improved by using GPUs, mainly due to the huge amount of parallel computations in large-scale neural networks. However, during the training of our model, we observed that the clustering of document representations become an efficiency bottleneck, even after we deploy a K-Means algorithm that runs on GPU. To solve this issue, we consider an asynchronous document encoding and clustering approach depicted in Figure 1b. In this training approach, we use two GPUs: we save a snapshot of the encoder parameters (i.e., ϕ) at the beginning of each training step,[4] and compute the document representations for all documents retrieved in response to all training queries. We then use the obtained cluster centroids (M_is) for training the model on the second GPU. While the model is being trained, the first GPU computes the document representations and cluster centroids for the next step. In fact, this approach may not be as effective as synchronous training, because the cluster centroids at each training step is obtained from the model parameters at two previous steps (i.e., as shown in Figure 1b, the model parameters from step $s-1$ produces the clusters for step $s+1$). However, the efficiency improvement provides enough incentives to consider asynchronous training. We do not have effectiveness comparison between the synchronous and asynchronous training strategies, as training the synchronous model would be impractical on a large dataset.

Training data and setup. Another challenge in training NMIR is related to its training data and especially ground truth intent descriptions. There are multiple ways of automatically creating training data for weak supervision training of the model, for example using query reformulation data or anchor text. In our experiments, we follow a weak supervision solution based on the MIMIC-Click dataset, recently released by Zamani et al. [81].[5] The authors extracted and generated the query intent descriptions by mining and predicting them from the Bing's search query logs. In more detail, the data is created based on query reformulation data with the goal of finding query reformulations that reveal different intents of the query. Since users mostly clarify their intents by adding one or more terms to their original query in a search session, often called query specialization [36], query intents can be predicted by extracting a set of query reformulation triples (q,qq',c) (or $(q,q'q,c)$), which denotes that the query q is followed by the query qq' (or $q'q$) in the same search session (i.e., immediate successive queries) with a frequency of c, when it is aggregated over the whole query log data for all users. qq' is the concatenation of q and q', where $|q'| > 0$. Since the mined query reformulations may refer to the same intent, a diversification based approach is used for identifying a diverse set of query intent descriptions [80]. The data consists of over 400,000 unique search queries and 2-5 intent descriptions per query.

In more detail, we use 80% of the MIMICS-Click queries for training and the rest for validation. The validation set is used for hyper-parameter tuning and early stopping. For the top retrieved documents (i.e., D_is), we used the SERP information fetched from the Bing's public web search API by the creators of the MIMICS dataset.[6] In our experiments, we use the document snippets as an accurate textual representation of the retrieved documents.

We used Adam optimizer with a batch size of 8 to train our model. The small batch size was selected due to the GPU memory constraints. We used early stopping based on the loss value on the validation set. The number of Guided Transformer layers was set to three. The learning rate was selected based on the validation loss from the $[1e-6, 5e-5]$ interval. We report the generated facets for a few example queries by NMIR in Table 3. The first part of the table includes some examples that the model successfully identified the facets of the query, and the second part include two failed queries. In the first failed query, the model could not distinguish between the word "window" and the windows operating system. As a result, it generated meaningless facet descriptions. The second failed query contains some facet descriptions that may be semantically related to the query, but are not coherent. One of the generated facets for this query is even very long and grammatically incorrect.

4 POTENTIAL APPLICATIONS OF NMIR

NMIR is a general framework with multiple applications in a wide range of IR tasks. For instance, NMIR can be simply used for **abstractive query intent (or facet) generation**. We use this task in our experiments to demonstrate the quality of learned representations. Another potential application of NMIR would be on **search result diversification**, as multiple query intent representations can help diversify a result list. One can imagine a clear application of NMIR in **exploratory search** tasks, where different representations of the search query can be used by the user to navigate through various aspects of the topic. In **conversational search**, asking clarifying questions has been recognized as an important and challenging task [2]. Multiple query representations can be used for generating and selecting clarifying questions in conversational search settings, that is also used in our experiments.

[4] Note that each training step includes 10000 batches in our experiments.

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

[6] The MIMICS SERP data is available at http://ciir.cs.umass.edu/downloads/mimics-serp/MIMICS-BingAPI-results.zip.

Apart from query representation and its applications, the proposed solution can be potentially adopted for a variety of tasks related to document representation. For instance, according to the scope hypothesis [57], long documents often cover several different topics. Therefore, learning multiple representations for each document can be further investigated using the proposed framework. This will have applications in **document clustering and categorization**. We believe that learning multiple query and document representations together can potentially lead to improvement in **document ranking** too, as the model would be theoretically able to accurately find the closest query intent to the document.

One can even imagine applications of the proposed framework beyond text representation. For instance, in **collaborative recommender systems**, models learn a single representation for each user and item from user-item interaction signals. However, users may have multiple different interests and a single user representation vector may lead to information loss. The proposed framework can be potentially extended to recommender systems by learning a variable number of user representations based on different user interests. This would further lead to recommendation precision improvement. It can be also used for explaining each recommendation. Such technique would also enable users to select what profile representation would they prefer to be used for the next recommendation, or they can be selected automatically based on the user's situational context [63].

5 EXPERIMENTS

We *extrinsically* evaluate NMIR on two different IR tasks: query facet generation and clarification selection. Following previous work on search clarification [2, 80], search result diversification [4, 5], and facet generation [17, 32], we focus on multi-faceted queries.

5.1 Query Facet Generation

In our first set of experiments for evaluating the NMIR framework, we focus on query facet generation. The task is defined as generating a number of textual facet descriptions for a given query.

5.1.1 Evaluation Data. To evaluate this task, we use the MIMICS-Manual dataset [81]. This public dataset consists of 2464 unique web search queries sampled from the Bing query logs. The dataset contains between two and five facets for each query. The quality of each set of facets was manually assessed by three trained annotators. The quality labels are either Bad, Fair, and Good. In our experiments, we left out the Bad facet sets and considered the ones with either Fair or Good labels as our ground truth. Note that according to Zamani et al. [81], the Fair label still meets the quality criteria for being presented in a commercial web search engine. Although we find this a high-quality test collection for evaluating the performance of our model, we still present a small follow up experiment with manual annotation to highlight the improvements compared to the baselines with a higher confidence.

Note that we made sure that the intersection between the training and the test queries is empty. Similar to training, the top retrieved documents for each query in the test set was obtained from the Bing's Web Search API. For more information, see the training data details and training setup in Section 3.3.

5.1.2 Evaluation Metrics. To evaluate query facet generation models, we adopt four sets of evaluation metrics. (1) Term overlap metrics: these metrics have been previously used for evaluating query facet extraction models [30]. They include Term Precision (TP), Term Recall (TR), and Term F1-measure (TF). These metrics basically compute the precision, recall, and F1-measure for the set of terms generated by the model with respect to the terms appeared in the ground truth data. For more information about these metrics, refer to [30]. (2) Exact match metrics: similar to term overlap, this metric also focuses on exact text matching but at the facet level. In other words, these metrics compute the precision, recall, and F1-measure of generating the exact facet description appeared in the ground truth. (3) Set BLEU scores: BLEU [50] is a widely adopted metric for text generation tasks, e.g., machine translation. However, it is defined between a single candidate text and a set of references. In our task, we deal with comparing two sets of text, one set is different facet descriptions generated by the model (R) and the other one is different facet descriptions in the ground truth test set (G). To compute Set BLEU, we first generate all permutations of R and then choose R^* such that $R^* = \text{argmax}_{R' \in \text{perm}(R)} \frac{1}{M} \sum_{i=1}^{M} \text{BLEU-4}(R'_i, G_i)$, where the subscript i denotes the facet index and $M = \max(|G|, |R|)$. We then compute the Set BLEU scores using $\frac{1}{M} \sum_{i=1}^{M} \text{BLEU-n}(R^*_i, G_i)$ for different n-grams. (4) Set BERT-Score: BERT-Score [86] has been recently used to compute the semantic similarity of a candidate text and a set of reference texts using the BERT representations [14]. We define Set Bert-Score as $\frac{1}{M} \sum_{i=1}^{M} \text{BERT-Score}(R^*_i, G_i)$. We compute this mean performance for all precision, recall, and F1-measures computed by the BERT-Score model.

5.1.3 Results and Discussion. We use the following baseline methods in our experiment:

- QDist [73]: QDist is a retrieval model that first generates multiple query variations and reformulations of the submitted query and learns a distribution over queries for retrieval. Even though this approach is not implemented for facet generation, its query variations can be seen as different query intents and can be used as a baseline for our model.
- QFI and QFJ [32]: We use the state-of-the-art variation of the QFI and QFJ methods [32] that were developed for facet extraction in web search. As described in Section 2, they are based on graphical models that estimate the probability of a hidden variable for modeling the extraction probability of each facet term. We followed the implementation details provided by the authors and selected the parameters using the validation set described in Section 3.3.
- QDMiner [17]: This is a competitive baseline for facet extraction from text and html documents. It is a hybrid approach that integrates multiple solutions for query facet extraction.
- BART [39]: We fine-tuned BART based on our training data, where the query and the top retrieved documents are the BART inputs and a concatenation of all query facet descriptions separated using a special token are the BART target output for training. Sequence-to-sequence models, like BART, provide strong performance for reformulation and facet generation tasks [49].

We emphasize that the QFI and QFJ models are shown to outperform other existing query facet extraction models [32]. There exist many methods that use metadata or taxonomies to produce query facets, which are out of the scope of this paper. For all the baselines, we follow the same hyper-parameter selection approach as the proposed model. Note that the main goal of this experiment is to provide extrinsic evaluation for the quality of the learned query intent representations. Therefore, we do not intent to show that NMIR is the state-of-the-art approach for facet generation, instead

Table 1: Results for the query facet generation experiment. All the improvements observed by NMIR compared to all the baselines are statistically significant.

#		Term Overlap			Exact Match			Set BLEU				Set BERT-Score		
facets	**Model**	**Prec**	**Recall**	**F1**	**Prec**	**Recall**	**F1**	**1-gram**	**2-gram**	**3-gram**	**4-gram**	**Prec**	**Recall**	**F1**
2	QDist	0.1637	0.1888	0.1676	0.0048	0.0046	0.0050	0.3841	0.1648	0.0438	0.0158	0.7649	0.7938	0.7807
	QFI	0.1936	0.2202	0.2033	0.0068	0.0061	0.0062	0.4070	0.1692	0.0515	0.0178	0.8057	0.8057	0.8007
	QFJ	0.2111	0.2023	0.2029	0.0072	0.0077	0.0072	0.4192	0.1835	0.0478	0.0076	0.8115	0.8020	0.8011
	QDMiner	0.2546	0.2369	0.2468	0.0089	0.0088	0.0088	0.5091	0.1931	0.0538	0.0089	0.8216	0.8162	0.8109
	BART	0.4621	0.5018	0.4888	0.0512	0.0500	0.0508	0.6413	0.6063	0.5709	0.5381	0.8616	0.8528	0.8540
	NMIR	**0.5195**	**0.6068**	**0.5539**	**0.1025**	**0.1040**	**0.1031**	**0.7333**	**0.6762**	**0.6403**	**0.6050**	**0.9170**	**0.9071**	**0.9062**
3	QDist	0.0929	0.1157	0.0957	0.0049	0.0045	0.0043	0.3518	0.1447	0.0341	0.0065	0.7418	0.7862	0.7366
	QFI	0.1330	0.1361	0.1337	0.0054	0.0052	0.0051	0.3868	0.1637	0.0407	0.0167	0.7916	0.8004	0.7797
	QFJ	0.1604	0.1801	0.1678	0.0065	0.0061	0.0064	0.3844	0.1695	0.0459	0.0135	0.7853	0.8021	0.7798
	QDMiner	0.1676	0.2024	0.2022	0.0082	0.0100	0.0084	0.4371	0.2014	0.0510	0.0169	0.7899	0.8100	0.7870
	BART	0.3672	0.4650	0.4193	0.0436	0.0410	0.0414	0.6025	0.5531	0.5040	0.4621	0.8390	0.8311	0.8293
	NMIR	**0.4279**	**0.5327**	**0.4687**	**0.0739**	**0.0720**	**0.0720**	**0.6960**	**0.6336**	**0.5949**	**0.5593**	**0.8840**	**0.8976**	**0.8775**
4	QDist	0.1725	0.2437	0.1876	0.0047	0.0044	0.0046	0.3843	0.1710	0.0543	0.0214	0.7674	0.7688	0.7769
	QFI	0.1951	0.2638	0.2223	0.0068	0.0064	0.0065	0.4014	0.1874	0.0642	0.0231	0.8005	0.8072	0.7969
	QFJ	0.1777	0.1454	0.1503	0.0064	0.0058	0.0060	0.3977	0.1800	0.0571	0.0212	0.7925	0.8047	0.7897
	QDMiner	0.1894	0.1672	0.1987	0.0065	0.0073	0.0068	0.4862	0.2230	0.0633	0.0230	0.8044	0.8040	0.7991
	BART	0.3165	0.4515	0.3896	0.0343	0.0348	0.0345	0.5940	0.5376	0.4611	0.4159	0.8222	0.8206	0.8175
	NMIR	**0.3898**	**0.5072**	**0.4358**	**0.0685**	**0.0677**	**0.0681**	**0.6940**	**0.6292**	**0.5899**	**0.5543**	**0.8802**	**0.8978**	**0.8775**
5	QDist	0.1557	0.1593	0.1440	0.0023	0.0024	0.0023	0.3387	0.1048	0.0439	0.0176	0.7165	0.7802	0.7192
	QFI	0.1605	0.1941	0.1720	0.0058	0.0050	0.0050	0.3539	0.1524	0.0523	0.0203	0.7603	0.8127	0.7584
	QFJ	0.1767	0.1348	0.1451	0.0055	0.0057	0.0053	0.3735	0.1675	0.0564	0.0234	0.7731	0.8136	0.7714
	QDMiner	0.2176	0.1443	0.1773	0.0069	0.0066	0.0065	0.4275	0.1826	0.0657	0.0234	0.7758	0.8036	0.7792
	BART	0.3043	0.4124	0.3558	0.0282	0.0263	0.0275	0.5087	0.4406	0.3969	0.3445	0.7633	0.8017	0.7660
	NMIR	**0.3877**	**0.4559**	**0.4121**	**0.0613**	**0.0584**	**0.0596**	**0.6313**	**0.5628**	**0.5222**	**0.4871**	**0.8442**	**0.8870**	**0.8405**
variable	QDist	0.0969	0.1564	0.1195	0.0017	0.0023	0.0019	0.1999	0.1134	0.0360	0.0107	0.6772	0.6855	0.6100
	QFI	0.1461	0.1748	0.1571	0.0057	0.0061	0.0059	0.2763	0.1269	0.0421	0.0140	0.7069	0.7113	0.6144
	QFJ	0.1807	0.2041	0.1894	0.0069	0.0067	0.0067	0.2484	0.1065	0.0242	0.0090	0.7196	0.6708	0.5871
	QDMiner	0.2060	0.2456	0.1894	0.0076	0.0083	0.0079	0.2893	0.1226	0.0301	0.0126	0.7220	0.7025	0.6285
	BART	0.4307	0.4618	0.4481	0.0474	0.0516	0.0486	0.4459	0.4003	0.3896	0.3351	0.7623	0.6932	0.6558
	NMIR	**0.4851**	**0.5673**	**0.4968**	**0.0790**	**0.0842**	**0.0784**	**0.5187**	**0.4748**	**0.4470**	**0.4192**	**0.8003**	**0.7487**	**0.6928**

Table 2: Manual annotation results for pairwise comparison of NMIR vs. BART in facet generation.

Win	Tie	Loss
48%	30%	22%

the goal is to demonstrate the quality of the learned representations through facet generation tasks.

The results are presented in Table 1. First, we observe that the proposed model consistently outperforms both probabilistic and neural baselines. This is true for all the evaluation metrics used in our experiment, including term matching, facet matching, n-gram matching, and semantic matching metrics. Note that all the improvements are statistically significant, according to the paired t-test with Bonferroni correction at 95% confidence.

We note that the test set for different number of facets is different. In other words, the numbers in different parts of Table 1 separated by a solid line should not be compared as their test queries are different. That being said, we still observe a consistent drop in the performance as the number of facets increases, which makes sense considering the fact that it becomes increasingly more difficult.

Another observation is the large performance gap between QDist, QFI, QFJ, and QDMiner with the neural models (BART and NMIR). The reason is that the former are extractive facet generation models, while the latter are abstractive generation models. The ground truth contains several terms for describing the facets that are not in the result list, thus the extractive models fall short in generating them. This explains the poor performance of the extractive models.

The next observation from the result table is that the Exact Match performances are substantially lower than the other metrics. Exact Match is an extremely strict metric that only focuses on generating the exact facet description text used in the ground truth. Term Overlap and Set BLEU provide smoother versions of term and phrase matching measures.

Furthermore, the results obtained by NMIR shows that it achieves higher Term Overlap Recall than Precision, and this is consistent across all the test sets. This shows that the percentage of generated terms not included in the ground truth is larger than those in the ground truth missed by the model. Moreover, we observe that the

Table 3: Some successful and unsuccessful examples of the facets generated by NMIR. Facets are separated using the ■ symbol.

Query	Generated facets
atropine sulfate drops	atropine sulfate drops interactions ■ atropine sulfate drops overdose ■ atropine sulfate drops precautions ■ atropine sulfate drops side effects ■ atropine sulfate drops uses
best fps games	best fps games for steam ■ best fps games for pc ■ best fps games for ps4 ■ best fps games for xbox one ■ best fps games for android
accident investigation	motorcycle accident ■ car accident ■ train accident ■ boat accident ■ forklift accident
awning window	awning window windows 10 ■ awned window windows 7 ■ a roofing window window windows 8 ■ a window windows vista ■ windows xp
balance of payment	balance of payment bank of america ■ balance of payments bank of ireland bank of germany bank of usa ■ bank of europe bank of philippines ■ cash balance

performance of non-parametric NMIR for the variable facet number case is closer to its performance when the number of generated facets are equal to 2. The main reason is that the number of queries with 2-3 facets are dominated in the MIMICS-Manual dataset.

We further extend our evaluation using manual annotation. We showed a query to the annotators and asked them review multiple pages of the result list for each query using a web search engine to understand different aspects of each query. We then showed them the facet descriptions generated by BART (our strongest baseline) and NMIR for the query and asked them to decide which one is a better facet description set, with respect to both quality and coverage. They could select one of them or vote for a tie. The presentation order (BART vs. NMIR) was random to reduce biases. We repeat this process for 100 queries randomly sampled from the test set by two annotators. In case of disagreement, we asked them to discuss and come up with an agreement or discard the query. The results for NMIR vs. BART are presented in Table 2. NMIR wins in 48% of the cases and loses in 22% of queries.

We made sure that each and every component in the proposed framework significantly contributes to the model's performance, however, due to space constraints, we drop the **ablation study** from the experimental results. Upon acceptance, we will include these results in an extended version of the paper on arXiv.

5.2 Clarifying Question Selection

Imagine a conversational search scenario, when users are allowed to seek their information need through natural language conversation. In case of ambiguous or faceted queries, the system would be allowed to ask clarifying questions to achieve a clear understanding of the user's information need.

The task of selecting next clarifying question, as has been described by the authors in [2] is selecting a proper clarifying question from a pool of questions given a user-system conversation. Similar to [2, 23], we evaluate the task based on the retrieval performance after asking the clarifying question(s).

We believe that learning multiple query representations would improve the task of selecting clarifying question as it provides an accurate representation of each intent that may need to be clarified. Therefore, the clarification selection task is used for extrinsic evaluation of the proposed model.

5.2.1 Data. To evaluate this task, we used the Qulac dataset [2], which has been constructed for search clarification in open-domain information seeking conversations. The queries in the dataset were borrowed from in the TREC Web Track 2009-2012 [6]. Therefore, Qulac contains 200 topics (two of which are omitted due to lack of relevance judgment). The queries were marked as either "ambigous" or "faceted" by the TREC Web Track organizers. The facets associated with each query and their relevance judgement are also given. After obtaining this information, the authors collected a number of clarifying question and their answers through multiple rounds of crowdsourcing. The average facets per topic is 3.85 ± 1.05 and Qulac contains a total of 10,277 question-answer pairs. We refer the reader to [2] for more information about the Qulac dataset.

5.2.2 Experimental Setup. For selecting clarifying question we re-rank all the clarifying questions in the pool with respect to their similarity to the conversation history up to current point. To elaborate more, we obtain multiple representations for the user query q (or the conversation history up to the current turn) with NMIR. For the top retrieved documents snippet that NMIR requires for its input, we use the top 10 documents retrieved from the ClueWeb09-Category B, using the query likelihood retrieval model [54] with Dirichlet prior smoothing [83]. The smoothing parameter μ was set to the average document length in the collection. For document indexing and retrieval, we use the open-source Galago search engine.[7] The spam documents were automatically identified and removed from the index using the Waterloo spam scorer[8] [7] with the threshold of 70%. Then, we apply the Indri snippet generation function to obtain document snippets. The clarifying questions are represented with a standard BART Encoder. In the last step, each representation of the query q is concatenated with the clarifying questions representations and is fed to a fully connected layer that generates the similarity score. The question with the highest similarity score is selected. The models are trained and evaluated using 5-fold cross validation over topics.

To be consistent with the experiments reported in [2], we consider up to three turns of conversations in the data, and report the average performance on retrieval after asking the clarifying question. Considering the focus of the task is selecting the proper clarifying question, we use query likelihood as the follow up retrieval model with the setting explained earlier. All the experimental settings in this section are consistent with the work of Aliannejadi et al. [2] who introduced the dataset. The hyper-parameters of the model, such as learning rate and batch size, are selected based on the detailed provided in Section 3.3.

5.2.3 Evaluation Metrics. Following the literature [2, 23], we evaluate the task of clarifying question selection based on the retrieval performance after the clarifying question has been asked. The rationale behind this is if a clarifying question is selected properly,

[7] http://lemurproject.org/galago.php

[8] https://plg.uwaterloo.ca/~gvcormac/clueweb09spam/

it should address the user's information need, and thus improve the search quality. Considering the nature of conversational search tasks, we focus on precision-oriented metrics for this experiment. We use standard IR metrics such as mean reciprocal rank (MRR), and normalized discounted cumulative gain (nDCG) [27] with ranking cut-offs of @1, @5, and @20. We report the average performance across different conversations in the data. The statistical significant improvements is computed using the paired t-test with Bonferroni correction at 95% confidence intervals (i.e., p-value < 0.05).

5.2.4 Results and Discussion. To evaluate our model, we compare its performance on the Qulac dataset with the following baselines:

- OriginalQuery: This shows the retrieval performance before asking any clarifying questions. This baseline shows how much improvement we obtain by asking a clarification.
- σ-QPP: We use a simple yet effective query performance predictor, σ [53], as an estimation of the question's quality. For a candidate clarifying question, we perform retrieval (without the answer) and estimate the ranking performance using σ. The clarifying question that leads to the highest σ is selected.
- LambdaMART and RankNet: These models re-rank clarifying questions based on a set of features, ranging from the query performance prediction to question templates to BERT similarities. The exact definition of feature descriptions can be found in [2].
- BERT-NeuQS: A model based on BERT used for clarifying question re-ranking proposed in [2]. The model concatenates the query, the conversation history, and the candidate clarifying question and feeds it to BERT. The obtained representation is then concatenated with some features, e.g., σ-QPP, for producing a single score for the candidate clarifying question.
- BERT-GT: The model proposed by Hashemi et al. [23] for clarification selection using BERT representations and incorporating the top retrieved documents through Guided Transformer. We used the single-task learning variation to have a fair comparison. The multi-task version uses the information not available to the other baselines and the proposed model.
- BART: We used the same BART model trained for facet generation in our last experiment (with k=5 and non-parametric) and further fine-tune its Encoder using the Qulac data, similar to the BERT-NeuQS model.

All the baselines and the proposed model are trained and evaluated using the same procedure as suggested by the authors of Qulac [2]. The results are reported in Table 4. The proposed solution led to significant improvement compared to all the baselines across all the metrics, except for nDCG@20. We also include the oracle lower-bound and upper-bound performances to the table to provide an insight on rooms available for improvement on this data. The results obtained by Oracle-Best Question show the tight gap with the upper-bound performance and the model performance, which explains the small improvements with respect to nDCG@20. Moreover, the results suggest that the non-parametric NMIR model outperforms the one with fixed number of clusters ($k = 5$). This might be due to the ability of the non-parametric model to generate fewer but more accurate representations for some queries.

Table 4: Results for the next clarifying question selection task, up to 3 conversation turns. * indicates statistically significant improvements compared to all the baselines with 95% confidence interval.

Method	MRR	nDCG@1	nDCG@5	nDCG@20
OriginalQuery	0.2715	0.1381	0.1451	0.1470
σ-QPP	0.3570	0.1960	0.1938	0.1812
LambdaMART	0.3558	0.1945	0.1940	0.1796
RankNet	0.3573	0.1979	0.1943	0.1804
BERT-NeuQS	0.3625	0.2064	0.2013	0.1862
BERT-GT	0.3784	0.2279	0.2107	0.1890
BART	0.3661	0.2083	0.2049	0.1891
NMIR k=5	0.3753	0.2211	0.2194*	0.1903
NMIR non-param	**0.3826***	**0.2327***	**0.2298***	**0.1920**
Oracle-Worst Question	0.2479	0.1075	0.1402	0.1483
Oracle-Best Question	0.4673	0.3031	0.2410	0.2077

6 CONCLUSIONS AND FUTURE WORK

In this paper, we introduced NMIR, a general framework that given a suitable resource is able to map one input sequence to widely distributed representations. NMIR learns multiple representations for each query to better represent faceted and ambiguous queries. We implemented the proposed framework using the state-of-the-art neural network architectures, such as BART for initializing the encoder and decoder parameters and Guided Transformer for mapping a generic query representation to an intent representation space. We also introduced an asynchronous optimization approach for efficient training of the framework. Our evaluation on query facet generation and search clarification selection tasks demonstrated the effectiveness of the proposed solution compared to competitive baselines. The NMIR framework has a wide range of applications in IR and NLP. In the future, we intend to extend the applications of the proposed framework to other major IR tasks, including document representation learning, search result diversification and relevance ranking. We will explore the potential of extending the proposed framework to other domains, e.g., learning multiple user representations in collaborative recommender systems and learning multiple representations for each node in a heterogeneous graph.

ACKNOWLEDGEMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF IIS-1715095. 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] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying Search Results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM '09)*. 5–14.

[2] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. ACM, New York, NY, USA, 475–484.

[3] Fei Cai and Maarten de Rijke. 2016. A Survey of Query Auto Completion in Information Retrieval. *Found. Trends Inf. Retr.* 10, 4 (2016), 273–363.

[4] 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)*. ACM, New York, NY, USA, 335–336.

[5] Ben Carterette and Praveen Chandar. 2009. Probabilistic Models of Ranking Novel Documents for Faceted Topic Retrieval. In *Proceedings of the 18th ACM Conference*

on Information and Knowledge Management (Hong Kong, China) *(CIKM '09)*. ACM, New York, NY, USA, 1287–1296.

[6] Charles L.A. Clarke, Nick Craswell, and Ian Soboroff. 2009. Overview of the TREC 2009 Web Track. In *Proceedings of the 2009 Text Retrieval Conference (TREC '09)*.

[7] Gordon V. Cormack, Mark D. Smucker, and Charles L. Clarke. 2011. Efficient and Effective Spam Filtering and Re-Ranking for Large Web Datasets. *Inf. Retr.* 14, 5 (2011), 441–465.

[8] W. B. Croft and D. J. Harper. 1979. Using Probabilistic Models of Document Retrieval Without Relevance Information. *J. of Documentation* 35, 4 (1979), 285–295.

[9] 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.

[10] Van Dang, Xiaobing Xue, and W. Bruce Croft. 2011. Inferring Query Aspects from Reformulations Using Clustering. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM '11)*. 2117–2120.

[11] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by Latent Semantic Analysis. 41, 6 (1990), 391–407.

[12] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to Attend, Copy, and Generate for Session-Based Query Suggestion. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (Singapore, Singapore) *(CIKM '17)*. ACM, New York, NY, USA, 1747–1756.

[13] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. ACM, New York, NY, USA, 65–74.

[14] 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 (NAACL '19)*. ACL, Minneapolis, Minnesota, 4171–4186.

[15] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query Expansion with Locally-Trained Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL '16)*. ACL, Berlin, Germany, 367–377.

[16] Zhicheng Dou, Sha Hu, Yulong Luo, Ruihua Song, and Ji-Rong Wen. 2011. Finding Dimensions for Queries. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM '11)*. 1311–1320.

[17] Zhicheng Dou, Zhengbao Jiang, Sha Hu, Ji-Rong Wen, and Ruihua Song. 2016. Automatically Mining Facets for Queries from Their Search Results. *IEEE Trans. on Knowl. and Data Eng.* 28, 2 (2016), 385–397.

[18] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. 1987. The Vocabulary Problem in Human-System Communication. *Commun. ACM* 30, 11 (Nov. 1987), 964–971.

[19] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth J.F. Jones. 2015. Word Embedding Based Generalized Language Model for Information Retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Santiago, Chile) *(SIGIR '15)*. ACM, New York, NY, USA, 795–798.

[20] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-Hoc Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (Indianapolis, Indiana, USA) *(CIKM '16)*. ACM, New York, NY, USA, 55–64.

[21] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. 2020. A Deep Look into neural ranking models for information retrieval. *Information Processing & Management* 57, 6 (2020), 102067.

[22] JA Hartigan and MA Wong. 1979. Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics* (1979), 100–108.

[23] Helia Hashemi, Hamed Zamani, and W. Bruce Croft. 2020. Guided Transformer: Leveraging Multiple External Sources for Representation Learning in Conversational Search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 1131–1140.

[24] Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. 2016. Learning to Rewrite Queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (Indianapolis, Indiana, USA) *(CIKM '16)*. ACM, New York, NY, USA, 1443–1452.

[25] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management* (San Francisco, California, USA) *(CIKM '13)*. ACM, New York, NY, USA, 2333–2338.

[26] N. Jardine and C.J. van Rijsbergen. 1971. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval* 7, 5 (1971), 217–240.

[27] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* 20, 4 (2002), 422–446.

[28] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 39–48.

[29] Christian Kohlschütter, Paul-Alexandru Chirita, and Wolfgang Nejdl. 2006. Using Link Analysis to Identify Aspects in Faceted Web Search.

[30] 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)*. ACM, New York, NY, USA, 93–102.

[31] Weize Kong and James Allan. 2014. Extending Faceted Search to the General Web. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management* (Shanghai, China) *(CIKM '14)*. 839–848.

[32] Weize Kong and James Allan. 2016. Precision-Oriented Query Facet Extraction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16)*. 1433–1442.

[33] Oren Kurland and Lillian Lee. 2009. Clusters, Language Models, and Ad Hoc Information Retrieval. *ACM Trans. Inf. Syst.* 27, 3, Article 13 (May 2009), 39 pages.

[34] Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query Expansion Using Word Embeddings. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (Indianapolis, Indiana, USA) *(CIKM '16)*. ACM, New York, NY, USA, 1929–1932.

[35] K. Latha, K. R. Veni, and R. Rajaram. 2010. AFGF: An Automatic Facet Generation Framework for Document Retrieval. In *2010 International Conference on Advances in Computer Engineering*. 110–114.

[36] Tessa Lau and Eric Horvitz. 1999. Patterns of Search: Analyzing and Modeling Web Query Refinement. In *Proceedings of the Seventh International Conference on User Modeling* (Banff, Canada) *(UM '99)*. Springer-Verlag, Berlin, Heidelberg, 119–128.

[37] Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New Orleans, Louisiana, USA) *(SIGIR '01)*. ACM, New York, NY, USA, 120–127.

[38] Anton V Leouski and W Bruce Croft. 2005. *An evaluation of techniques for clustering search results*. Technical Report. Massachusetts Univ Amherst Dept of Computer Science.

[39] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 7871–7880.

[40] Chengkai Li, Ning Yan, Senjuti B. Roy, Lekhendro Lisham, and Gautam Das. 2010. Facetedpedia: Dynamic Generation of Query-Dependent Faceted Interfaces for Wikipedia. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. 651–660.

[41] Jiwei Li and Dan Jurafsky. 2015. Do Multi-Sense Embeddings Improve Natural Language Understanding?. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. ACL, Lisbon, Portugal, 1722–1732.

[42] Ying Li, Zijian Zheng, and Honghua (Kathy) Dai. 2005. KDD CUP-2005 Report: Facing a Great Challenge. *SIGKDD Explor. Newsl.* 7, 2 (Dec. 2005), 91–99.

[43] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2020. Pretrained Transformers for Text Ranking: BERT and Beyond. *arXiv preprint arXiv:2010.06467* (2020).

[44] Xiaoyong Liu and W. Bruce Croft. 2004. Cluster-Based Retrieval Using Language Models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Sheffield, United Kingdom) *(SIGIR '04)*. ACM, New York, NY, USA, 186–193.

[45] A. Meyerson. 2001. Online facility location. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. 426–431.

[46] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (Lake Tahoe, Nevada) *(NeurIPS '13)*. Curran Associates Inc., Red Hook, NY, USA, 3111–3119.

[47] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match Using Local and Distributed Representations of Text for Web Search. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1291–1299.

[48] Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2015. Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space. *CoRR* (2015).

[49] Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-Oriented Query Reformulation with Reinforcement Learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. ACL, Copenhagen, Denmark, 574–583.

[50] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (Philadelphia, Pennsylvania) *(ACL '02)*. ACL, USA, 311–318.

[51] Dae Hoon Park and Rikio Chiba. 2017. A Neural Language Model for Query Auto-Completion. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. ACM, New York, NY, USA, 1189–1192.

[52] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP '14)*. ACL, Doha, Qatar, 1532–1543.

[53] Joaquín Pérez-Iglesias and Lourdes Araujo. 2010. Standard Deviation as a Query Hardness Estimator. In *SPIRE '10* (Los Cabos, Mexico).

[54] Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Melbourne, Australia) *(SIGIR '98)*. ACM, New York, NY, USA, 275–281.

[55] Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. *Open-Retrieval Conversational Question Answering*. ACM, New York, NY, USA, 539–548.

[56] Filip Radlinski and Susan Dumais. 2006. Improving Personalized Web Search Using Result Diversification. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06)*. 691–692.

[57] Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

[58] J. J. Rocchio. 1971. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*. 313–323.

[59] G. Salton, A. Wong, and C. S. Yang. 1975. A Vector Space Model for Automatic Indexing. *Commun. ACM* 18, 11 (Nov. 1975), 613–620.

[60] Mark Sanderson. 2008. Ambiguous Queries: Test Collections Need More Sense. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Singapore, Singapore) *(SIGIR '08)*. ACM, New York, NY, USA, 499–506.

[61] Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting Query Reformulations for Web Search Result Diversification. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*. 881–890.

[62] Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. 2015. Search Result Diversification. *Found. Trends Inf. Retr.* 9, 1 (March 2015), 1–90.

[63] Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo, and Mehdi Elahi. 2018. Current challenges and visions in music recommender systems research. *Int. J. Multim. Inf. Retr.* 7, 2 (2018), 95–116.

[64] Emilia Stoica, Marti Hearst, and Megan Richardson. 2007. Automating Creation of Hierarchical Faceted Metadata Structures. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*. 244–251.

[65] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (Montreal, Canada) *(NIPS'14)*. MIT Press, Cambridge, MA, USA, 3104–3112.

[66] Jaime Teevan, Susan Dumais, and Zachary Gutt. 2008. Challenges for Supporting Faceted Search in Large, Heterogeneous Corpora like the Web. In *HCIR 2008*.

[67] Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2018. Diverse Beam Search for Improved Description of Complex Scenes. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Press, 7371–7379.

[68] Ellen M. Voorhees. 1985. The Cluster Hypothesis Revisited. In *Proceedings of the 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Montreal, Quebec, Canada) *(SIGIR '85)*. ACM, New York, NY, USA, 188–196.

[69] Sida Wang, Weiwei Guo, Huiji Gao, and Bo Long. 2020. *Efficient Neural Query Auto Completion*. ACM, New York, NY, USA, 2797–2804.

[70] Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural Text Generation With Unlikelihood Training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=SJeYe0NtvH

[71] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR* abs/1910.03771 (2019). arXiv:1910.03771 http://arxiv.org/abs/1910.03771

[72] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *International Conference on Learning Representations (ICLR '21)*.

[73] Xiaobing Xue and W. Bruce Croft. 2013. Modeling Reformulation Using Query Distributions. *ACM Trans. Inf. Syst.*, Article 6 (May 2013), 34 pages.

[74] Liu Yang, Minghui Qiu, Chen Qu, Cen Chen, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, and Haiqing Chen. 2020. *IART: Intent-Aware Response Ranking with Transformers in Information-Seeking Conversation Systems*. ACM, New York, NY, USA, 2592–2598.

[75] Hamed Zamani, Michael Bendersky, Xuanhui Wang, and Mingyang Zhang. 2017. Situational Context for Ranking in Personal Search. In *Proceedings of the 26th International Conference on World Wide Web* (Perth, Australia) *(WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1531–1540.

[76] Hamed Zamani and W. Bruce Croft. 2016. Embedding-Based Query Language Models. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval* (Newark, Delaware, USA) *(ICTIR '16)*. ACM, New York, NY, USA, 147–156.

[77] Hamed Zamani and W. Bruce Croft. 2016. Estimating Embedding Vectors for Queries. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval* (Newark, Delaware, USA) *(ICTIR '16)*. ACM, New York, NY, USA, 123–132.

[78] Hamed Zamani and W. Bruce Croft. 2017. Relevance-Based Word Embedding. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. ACM, New York, NY, USA, 505–514.

[79] Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From Neural Re-Ranking to Neural Ranking: Learning a Sparse Representation for Inverted Indexing. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. ACM, New York, NY, USA, 497–506.

[80] Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating Clarifying Questions for Information Retrieval. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. ACM, New York, NY, USA, 418–428.

[81] Hamed Zamani, Gord Lueck, Everest Chen, Rodolfo Quispe, Flint Luu, and Nick Craswell. 2020. MIMICS: A Large-Scale Data Collection for Search Clarification. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. ACM, New York, NY, USA, 3189–3196.

[82] Hamed Zamani, Bhaskar Mitra, Everest Chen, Gord Lueck, Fernando Diaz, Paul N. Bennett, Nick Craswell, and Susan T. Dumais. 2020. *Analyzing and Learning from User Interactions for Search Clarification*. ACM, New York, NY, USA, 1181–1190.

[83] Chengxiang Zhai and John Lafferty. 2001. Model-Based Feedback in the Language Modeling Approach to Information Retrieval. In *Proceedings of the Tenth International Conference on Information and Knowledge Management* (Atlanta, Georgia, USA) *(CIKM '01)*. ACM, New York, NY, USA, 403–410.

[84] Cheng Xiang Zhai, William W. Cohen, and John Lafferty. 2003. Beyond Independent Relevance: Methods and Evaluation Metrics for Subtopic Retrieval. ACM, New York, NY, USA, 10–17.

[85] Hongfei Zhang, Xia Song, Chenyan Xiong, Corby Rosset, Paul N. Bennett, Nick Craswell, and Saurabh Tiwary. 2019. Generic Intent Representation in Web Search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. ACM, New York, NY, USA, 65–74.

[86] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=SkeHuCVFDr