Recognizing Medical Search Query Intent by Few-shot Learning

Yaqing Wang*
wangyaqing01@baidu.com
Baidu Research
Baidu Inc.
Beijing, China

Song Wang*
wangsong162768@gmail.com
Baidu Research & Department of ECE
Baidu Inc. & University of Virginia
Beijing, China

Yanyan Li
liyanyanliyanyan@baidu.com
Baidu Research
Baidu Inc.
Beijing, China

Dejing Dou
doudejing@baidu.com
Baidu Research
Baidu Inc.
Beijing, China

ABSTRACT

Online healthcare services can provide unlimited and in-time medical information to users, which promotes social goods and breaks the barriers of locations. However, understanding the user intents behind the medical related queries is a challenging problem. Medical search queries are usually short and noisy, lack strict syntactic structure, and also require professional background to understand the medical terms. The medical intents are fine-grained, making them hard to recognize. In addition, many intents only have a few labeled data. To handle these problems, we propose a few-shot learning method for medical search query intent recognition called MEDIC. We extract co-click queries from user search logs as weak supervision to compensate for the lack of labeled data. We also design a new query encoder which learns to represent queries as a combination of semantic knowledge recorded in an external medical knowledge graph, syntactic knowledge which marks the grammatical role of each word in the query, and generic knowledge which is captured by language models pretrained from large-scale text corpus. Experimental results on a real medical search query intent recognition dataset validate the effectiveness of MEDIC.

CCS CONCEPTS

• **Information systems** → **Query representation**; **Query intent**; *Query log analysis*; • **Computing methodologies** → **Supervised learning by classification**.

KEYWORDS

Medical Search Query Understanding; Query Intent Recognition; Co-click Query Analysis; Few-shot Learning; Knowledge Graph; Graph Representation Learning

*Equal contribution. S. Wang did his work during internship at Baidu Research.

SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531789

ACM Reference Format:
Yaqing Wang, Song Wang, Yanyan Li, and Dejing Dou. 2022. Recognizing Medical Search Query Intent by Few-shot Learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3531789

1 INTRODUCTION

Online healthcare services allow users to easily reach medical and healthcare related knowledge via web search. They break the limitation of time and space and also reduce the medical consultation cost. Besides, users can freely express privacy issues such as venereal disease without feeling embarrassed. It is reported that around 7 percent of Google's daily searches are health-related [29], and more than 200 million Chinese search for medical knowledge on Baidu every day [45]. To understand medical queries properly, it is crucial to recognize the search intent such that satisfactory information and appropriate advertisements can be suggested [39].

Table 1: Examples of medical search queries. The blue italic words are entities existing in the medical KG.

Medical Search Query	Intent Label
I have *rhinitis*. Can I drink coffee?	Drug-Food
Stuffy nose which drug can I use.	Disease
Can I take *Astelin* and *Aspirin* together?	Drug-Drug
Have a stomachache after taking *Aspirin*.	Side Effect
Does *running nose* mean *COVID-19*?	Disease

Medical search query intent recognition faces two challenges:

Challenge 1. Medical search queries are particularly hard to understand. As search queries are provided by users, they usually consist of a few words which **lack context information**, **may not follow strict syntactic structure** of a written language and even contain typos [36, 39]. This is because users with different backgrounds can express the same intent by different words and expressions. Moreover, medical search queries **ask about professional medical terms** such as *rhinitis, Astelin and Aspirin* in Table 1. To bring in more semantic knowledge, several works [7, 15] put efforts on query conceptualization which learns to map search queries to concepts existing in external knowledge bases.

Challenge 2. Intent labels designed for medical search query have to be **fine-grained** to return **precise information** to the users [42]. Hence, many intents only **contain a few labeled queries** as shown in Figure 1. This is different from the general search query intent recognition problem, whose intents are usually grouped into informational, navigational, and transactional [4, 39, 43]. Thus, simple keyword matching or entity linking with respect to a medical knowledge graph (KG) [18] cannot recognize the intent of medical search queries. For examples, the first and the second queries in Table 1 mention similar diseases, but they have different search intents. Likewise, the third and the fourth queries both contain *Aspirin*, while their intents are quite different. Connecting *Aspirin* to entity type *drug* is not enough to recognize the intent. Moreover, due to the high requirement on domain knowledge, it is hard and expensive to hire appropriate annotators to label the data.

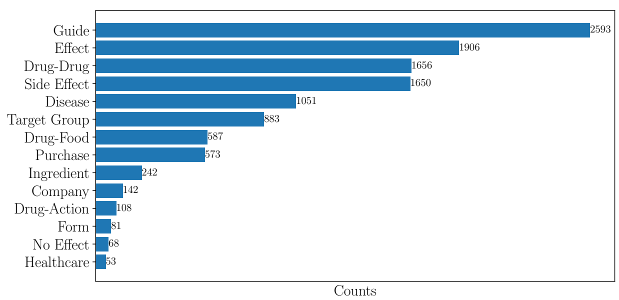


Figure 1: Intent distribution of our dataset.

Existing methods mainly work towards general query intent recognition [14, 43], and require large-scale labeled samples to obtain decent performance [27, 42]. How to train a model which can generalize to tasks with a limited number of labeled samples for medical search query intent recognition tasks is still an unresolved problem. To fill in this blank, we propose a few-shot learning approach for **MED**ical search query **I**ntent **C**lassification (**MEDIC**) which can leverage external semantic and syntactic knowledge, and discover co-click queries as weak supervision to compensate for the lack of enough labeled data. Specifically, our contribution can be summarized as follows:

- We propose a query encoder which simultaneously encodes external semantic knowledge from a medical KG, syntactic knowledge from Part-of-speech (POS) tags, and generic knowledge captured from large corpus. In particular, we leverage an external medical KG which contains professional medical knowledge, and construct a POS tag graph based on data-specific co-occurrence statistics to exploit syntactic knowledge. As the sets of entity types and POS tags are limited and can be easily covered by training data, the proposed model can be freely applied for new samples and new classes.
- Based on the insight that two queries which lead different users to click the same suggested URLs have high potential to express the same intent, we leverage co-click queries as weak supervision to augment the few labeled data. We specially design a class-aware contrastive loss to encourage the original labeled queries and co-click queries to obtain similar query embeddings, which can better exploit co-click queries as found in our empirical study.
- Extensive experiments are performed on a real medical search query intent recognition dataset collected from Baidu[1], which is one of the largest commercial search engine in the world. Results show that MEDIC consistently outperforms existing methods.

The rest of the paper is organized as follows. Section 2 introduces a real medical search query intent recognition dataset used in the paper. Section 4 presents the proposed method. Section 5 shows the experimental results. The last section concludes the paper with some future directions.

2 DATASET DESCRIPTION

The statistics of dataset used in this paper is in Table 2. All these data sources are in Chinese.

Table 2: Statistics of the dataset used in this paper.

Class	Number of classes	14
	Number of training classes	9
	Number of testing classes	5
Query	Number of queries	11593
	Number of co-click queries	165063
Medical KG	Number of medical entities	41676
	Number of medical entity types	20
POS Tag	Number of POS tag types	57

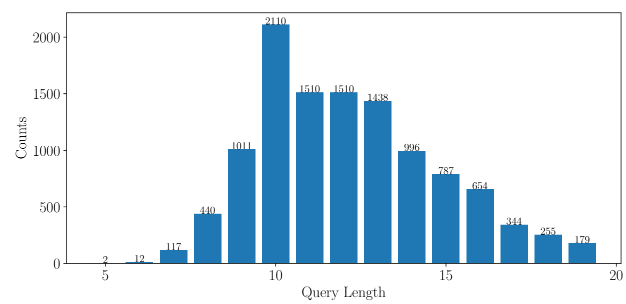


Figure 2: Query length distribution of our dataset.

Query Collection. We collect search queries from Baidu, one of the largest commercial search engines in the world, from December 2020 to March 2021. As we focus on drug-oriented intents, queries which do not contain any medical entity need to be dropped. To achieve this, we conduct entity linking of these queries with respect to a drug-oriented medical KG provided by Health IT Accelerator[2]. This KG integrates various data sources including the drug inserts, catalog of medicines covered by national medical insurance system

[1] https://www.baidu.com
[2] https://hita.omaha.org.cn

and Chinese Pharmacopoeia Commission. Afterwards, we record the entities and their corresponding entity types found in each query. We drop queries with length less than 5 as they carry too limited information.

Query Annotation. According to the search intent, each query is categorized into one of the 14 intents including (i) inquiries about drug itself: company, ingredient, purchase, and form; (ii) inquiries about the purpose of drug: disease and healthcare; (iii) inquiries about the usage of drug: guide and target group; (iv) inquires about the effect of drug: (normal) effect, side effect and no effect at all; and (v) inquiries about interactions with drug: drug-drug, drug-food, and drug-action. Those ambiguous queries which are hard to recognize their intents are marked as NA. We randomly sample 15000 queries and pay three well-educated annotators with bachelors' degrees to label the queries. The annotators first took a 3-hour tutorial provided by medical professionals. The annotated queries went through four rounds of quality assurance testing in five weeks. Only queries with at least two identical intent labels are kept. For these labeled queries, we collect co-click queries from user search logs during the same period. In addition, we assign each word a POS tag after tokenization via Jieba[3]. We present some labeled queries in Table 1, which are translated into English for ease of understanding. Figure 2 shows the query length distribution and Figure 1 shows the medical intent distribution of the queries.

Ethical Issue. No information about users who issued the queries is provided: the dataset contains labeled queries and their co-click queries, entities with their types and POS tags, which all are non-personally identifiable information.

3 RELATED WORKS

Search query intent recognition problem assigns predefined intent labels to search queries [32]. It is a typical type of text classification problem [24, 36]. Thus, we first provide a short review on text classification method. We then briefly review few-shot learning, which handles new tasks with a few labeled samples. Specially, we highlight the differences between existing works with our MEDIC.

3.1 Text Classification

Text classification is a fundamental task in natural language processing, which classifies documents into predefined taxonomy [24]. As deep neural networks such as convolutional neural networks (CNN) [20] and long short-term memory (LSTM) [26] can directly obtain expressive representations from raw texts in an end-to-end manner, many deep models are developed for text classification. Recently, graph neural networks (GNNs) [22] obtain state-of-the-art performance [37]. They can be divided into two types. The first type conducts transductive learning on a heterogeneous corpus-level graph which takes all text and word of the corpus as nodes and. It allows unlabeled texts to be labeled by propagating label information from neighboring texts via semi-supervised learning [41]. Another type models each document as a graph of word nodes and classifies the whole graph [10, 17], which allows inductive learning for new documents. However, this type of GNNs cannot effectively handle the case where only a few documents are labeled. Although we design our MEDIC upon GNN, we do not model the relationship between words nor documents. Instead, we construct two graphs to model the relationship among KG entity types and POS tags respectively, which can bring in external domain knowledge and syntactic information to better understand the medical queries.

Apart from models specially designed for text classification, pre-trained language models (PLMs) have demonstrated their ability of encoding universal knowledge, which can be taken as good starting point for downstream tasks. Examples include BERT [9] and GPT-3 [5]. A recent work infuses biomedical KG into BERT and evaluates on various English downstream tasks [27]. A direct way of leveraging these PLMs is to fine-tune them by taking gradient descent with respect to the objective of downstream tasks. However, when only a few labeled samples are provided, fine-tuning the PLMs has a high risk of overfitting. In our work, we also use a PLM which captures generic knowledge. But to appropriately solve the few-shot medical search query intent recognition problem, we have to incorporate external domain knowledge and manage to fully exploit provided data without overfitting.

3.2 Few-shot Learning

Few-shot learning (FSL) targets at generalizing to new tasks with a few labeled samples [38], which has been applied to text classification task. Several approaches conduct data augmentation to directly compensate for the lack of labeled samples. Examples include modifying original labeled samples with simple operations [40], selecting potential samples from unlabeled data [28] or synthesizing new samples in the feature space [35]. Another line of methods learns to embed samples into a space where samples can be easily discriminated [2]. Recently, there emerge few-shot learning methods developed for query intent detection in goal-oriented dialogue systems [12, 44], which is often jointly optimized with slot filling. In contrast, we target at recognizing the intent of medical queries drawn from online search engine. In addition, we augment the supervised information by exploiting co-click queries readily existing user search logs, which is able to bring in diverse semantic information.

4 METHODOLOGY

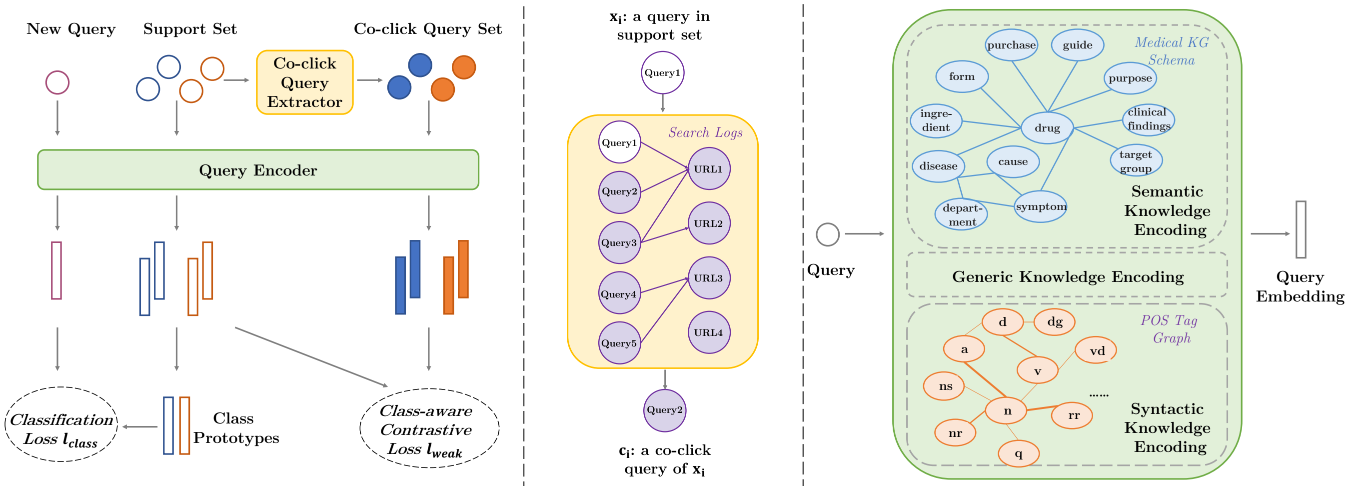
We now present the proposed MEDIC, whose high-level illustration is plotted in Figure 3. In the sequel, we first provide the formal problem formulation of few-shot medical search query classification tasks (Section 4.1). Then, we introduce the two key components of MEDIC: co-click query extractor which extracts co-click queries for each labeled query (Section 4.2), and query encoder which outputs query embeddings for queries (Section 4.3). Finally, we describe the training and inference procedure (Section 4.4).

4.1 Problem Formulation

In this paper, we denote scalars by lowercase, vectors by lowercase boldface, matrices by uppercase boldface, and sets by uppercase calligraphic font. For a vector $\mathbf{x}$, $[\mathbf{x}]_i$ denotes the ith element of $\mathbf{x}$. For a matrix $\mathbf{X}$, $[\mathbf{X}]_{ij}$ denotes the (i, j)th entry of $\mathbf{X}$.

We target at learning a predictor which generalize to recognize new fine-grained medical intents given a few exemplar labeled queries. Following the classic training protocol to handle tasks with

[3]https://github.com/fxsjy/jieba



(a) Overview. (b) Co-click Query Extractor. (c) Query Encoder.

Figure 3: A high-level illustration of the proposed MEDIC. We consider a 2-way 2-shot task $\mathcal{T}_t$ for clarity. For each query $(\mathbf{x}_i, y_i)$ in the support set $\mathcal{S}_t$, we use the co-click query extractor (Figure 3 (b)) to extract its co-click query which leads the users to click the same suggested URL returned by the search engine and is highly possible to express the same intent and can augment the supervised information. Then, we feed all queries in support set $\mathcal{S}_t$, class-wise co-click query sets $\{C_t^c\}_{c=1}^2$ and query set Q_t to the same query encoder to obtain their query embeddings. In particular, our query encoder (Figure 3 (a)) simultaneously encodes semantic knowledge provided by a medical KG schema $\mathcal{G}^s$, syntactic knowledge recorded in the POS tag graph $\mathcal{G}^g$, and the generic knowledge extracted from a large corpus into the query embeddings. MEDIC is optimized in an end-to-end manner with respect to the classification loss $\ell_{\text{class}}(Q_t)$ evaluated on the query set and the class-aware contrastive loss $\ell_{\text{class}}(\mathcal{S}_t)$ calculated using $\mathcal{S}_t$ and $\{C_t^c\}$.

a few labeled samples [2, 3, 12, 13], we employ episodic training to learn the predictor from a set of few-shot tasks $\{\mathcal{T}_t\}_{t=1}^{N_t}$. Each $\mathcal{T}_t$ is formulated as a N-way K-shot task, where a subset of N classes is randomly sampled from N_{train} training classes and K labeled samples $(\mathbf{x}_i, y_i)$ are then sampled per class. In our task, each $\mathbf{x}_i$ is a medical search query and y_i is its medical intent label. These $N \cdot K$ labeled samples form the support set $\mathcal{S}_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N \cdot K}$. The predictor will learn to predict the labels of samples in the query set $Q_t = \{(\mathbf{x}_j, y_j)\}$ given the few-shot $\mathcal{S}_t$. By learning from a set of different tasks, the model is expected to generalize to a new task $\hat{T}$ given a few labeled queries from N_{test} classes that are unseen during training.

Although this formulation takes the same form of few-shot text classification problem [2], the problem considered here is actually much harder: (i) a medical search query $\mathbf{x}_i$ is usually noisy, short and requires professional medical knowledge; and (ii) its intent y_i is fine-grained which requires a deep understanding of $\mathbf{x}_i$.

4.2 Co-click Query Extractor

When the number of labeled queries per class is limited, a natural consideration is to augment the supervised information. A straightforward way to conduct data augmentation via transforming original queries. However, due to the complicated syntactic structures, generating texts with target labels is much harder than images [8, 16]. Simple operations such as synonym replacement, random insertion/swap/deletion [40] may lead to performance gain, but they can easily change the original semantic meaning [35]. While it is expensive to consider conditional generation and round-trip translation to augment online search queries. Therefore, we instead turn to exploit co-click search queries readily recorded in user search logs and extract them via the proposed co-click query extractor.

Co-click search queries [1, 25, 43] of a search query refer to those search queries which lead the users to click the same suggested URL returned by the search engine. In Figure 3 (b), Query 1, Query 2 and Query 3 lead the users to click the URL 1. Thus, Query 2 and Query 3 are both co-click queries of Query 1. As the user intent lies under the click histories, if two users click the same URL, their search queries are very likely to express the same intent. In this way, we leverage the intelligence of users to detect those candidate queries which possibly express the same intent. Besides, users with different background can naturally express the same intent quite differently, such as using different words and tenses, describing in a formal or informal way. Therefore, leveraging these co-click search queries can bring in more diverse semantic information. For each $\mathbf{x}_i \in \mathcal{S}_t$, we extract one co-click search query $\mathbf{c}_i$. Then, we collectively put K co-click queries of class c into the class-wise co-click query set:

$$C_t^c = \{\mathbf{c}_i | \mathbf{c}_i \text{ is a co-click query of } \mathbf{x}_i \text{ and } y_i = c\}. \tag{1}$$

Intuitively, consider $\mathbf{x}_i, \mathbf{x}_m \in \mathcal{S}_t$, assume $\mathbf{c}_i$ is the co-click queries of $\mathbf{x}_i$ and and $\mathbf{c}_m$ is the co-click queries of $\mathbf{x}_m$. If the users do not click URLs at random, it is highly possible that $\mathbf{x}_i, \mathbf{x}_m, \mathbf{c}_i, \mathbf{c}_m$ describe the same intent. Therefore, we take C_t^c as a whole without discriminating the exact co-click relationship between $\mathbf{x}_i \in \mathcal{S}_t$ and $\mathbf{c}_i \in C_t^c$.

4.3 Query Encoder

Provided with $\mathcal{S}_t$, $\{C_t^c\}$ and Q_t, we describe how to obtain query embeddings via our specially designed query encoder. Specially, the resultant query embeddings simultaneously encode (i) semantic knowledge recorded in an external drug-oriented medical KG, (ii) syntactic knowledge which marks the grammatical role for each word in the query, and (iii) generic knowledge which is captured by language models pretrained from large-scale text corpus. Next, we first introduce how to encode the above-mentioned knowledge into fix-sized embeddings respectively, then show how to combine them into the query embeddings.

4.3.1 Semantic Knowledge Encoding. Medical search queries lack context information and usually contains professional medical entities, which makes them hard to understand. Therefore, we turn to external medical KGs to leverage their rich medical knowledge. As shown in Table 1, simply entity linking or keyword matching cannot reveal the medical intents. Queries containing the same medical entity can express quite different intents. Moreover, we find most queries only contain one recorded entity in our dataset. A single noun cannot reveal the intent. Nevertheless, these medical entities take an important semantic role in the queries, and also are the hardest to understand. Therefore, we use the drug-oriented medical KG introduced in Section 2. It contains comprehensive information for medical entities and relations to help alleviate the necessity of involving a human expert. However, we cannot directly use the medical KG. Except that the medical KG is large to operate, the resultant model cannot handle new entities and requires retraining. While in practice, it is hard to presume that training data and testing data contain the same set of entities. Inspired by [31] which founds the information of entity mention comes mainly from its type information, we instead use the medical KG schema. A schema is a generalized graph (or meta-level graph) of entity types connected by relation types [11]. The schema used in this paper is plotted in Figure 3 (c), which is provided along with the medical KG.

In order to leverage the semantic knowledge of the medical KG schema, we first learn to represent the relation types. Denote the medical KG schema as $\mathcal{G}^s = \{\mathcal{V}^s, \mathbf{A}^s\}$ where $\mathcal{V}^s$ is a set of entity types and $\mathbf{A}^s$ is the adjacency matrix. We initialize the node feature $\bar{\mathbf{h}}_i^s$ for $v_i^s \in \mathcal{V}^s$ as a one-hot vector. Formally, the (i, j)th entry $[\mathbf{A}^s]_{ij}$ of $\mathbf{A}^s$ is set as

$$[\mathbf{A}^s]_{ij} = \begin{cases} 1 & \text{if } v_i^s \text{ and } v_j^s \text{ are connected in } \mathcal{G}^s \\ 0 & \text{otherwise} \end{cases}.$$

In our $\mathcal{G}^s$, the connection between v_i^s and v_j^s means the corresponding two entity types are associated, such as *disease* and *cause*. While those disconnected are less likely to be semantically related, such as *form* and *department*. Let $\bar{\mathbf{H}}^s$ denote the initialized feature matrix for all nodes in $\mathcal{V}^s$, where the ith row of $\bar{\mathbf{H}}^s$ is $\bar{\mathbf{h}}_i^s$. We then use a 2-layer graph convolutional network (GCN) [22] to obtain the entity type embedding matrix $\mathbf{H}^s$ of $\mathcal{G}^s$:

$$\mathbf{H}^s = \tilde{\mathbf{A}}^s \cdot \mathrm{ReLU}(\tilde{\mathbf{A}}^s \bar{\mathbf{H}}^s \mathbf{W}_1^s)\mathbf{W}_2^s, \tag{2}$$

where $[\mathrm{ReLU}(\mathbf{x})]_i = \max([\mathbf{x}]_i, 0)$, $\tilde{\mathbf{A}}^s = \mathbf{D}^{s^{-\frac{1}{2}}}(\mathbf{I} + \mathbf{A}^s)\mathbf{D}^{s^{-\frac{1}{2}}}$ with $[\mathbf{D}^s]_{ii} = \sum_j [\mathbf{A}^s]_{ij}$, and $\mathbf{W}_1^s, \mathbf{W}_2^s$ are trainable parameters. In this way, the semantic knowledge of the medical KG is abstracted into $\mathbf{H}^s$.

Next, we describe how to represent queries in terms of $\mathbf{H}^s$. For each query, we conduct entity linking to identify the entities existing in the query and extract the corresponding entity types recorded in the medical KG. Formally, we use $\mathbf{s}_i^s$ to record the existence of each entity type v_j^s in query $\mathbf{x}_i$:

$$[\mathbf{s}_i^s]_j = \begin{cases} 1 & \text{if } v_j^s \text{ appears in } \mathbf{x}_i \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

Then, we represent a query $\mathbf{x}_i$ as a linear combination of node embeddings obtained in (2) as

$$\mathbf{e}_i^s = c^s \mathbf{H}^{s\top} \mathbf{s}_i^s, \tag{4}$$

where $c^s = 1/\|\mathbf{H}^{s\top}\mathbf{s}_i^s\|_2$ is a normalization scalar such that $\mathbf{e}_i^s$ has unit norm, and superscript $(\cdot)^\top$ denotes the transpose operation. In this way, $\mathbf{e}_i^s$ explains $\mathbf{x}_i$ from the perspective of semantic knowledge.

4.3.2 Syntactic Knowledge Encoding. In order to understand those medical search queries which lack strict syntactic structure and have grammatical errors, we further manage to encode the syntactic knowledge into query embeddings. This is achieved by leveraging POS tags which are commonly used to mark the syntactic role of each word with respect to the sentence and therefore can help discriminate ambiguous words. We first use a POS tagger to obtain the POS tag for each word of short text in $\mathcal{S}_t$, which forms the POS tag node set $\mathcal{V}^p$. Inspired by recent work [37], we construct the POS tag graph $\mathcal{G}^p = \{\mathcal{V}^p, \mathbf{A}^p\}$ (Figure 3 (c)) based on co-occurrence statistics calculated by point-wise mutual information (PMI):

$$[\mathbf{A}^p]_{ij} = \max(\mathrm{PMI}(v_i^p, v_j^p), 0).$$

In detail, $\mathrm{PMI}(v_i^p, v_j^p) = \log(p(v_i^p, v_j^p)/p(v_i^p)p(v_j^p))$, where $p(v_i^p)$ is the ratio of sliding windows that contain v_i^p over all the slide windows in $\mathcal{S}_t$, and $p(v_i^p v_j^p)$ is the ratio of sliding windows where v_i^p and v_j^p co-occur among all the sliding windows in $\mathcal{S}_t$. Denote the feature matrix for all nodes in $\mathcal{V}^p$ as $\bar{\mathbf{H}}^p$, where the ith row of $\bar{\mathbf{H}}^p$ is the feature for $v_i^p \in \mathcal{V}^p$ which is initialized as a one-hot vector. Similar to (2), we use a 2-layer GCN to obtain the POS tag embedding matrix $\mathbf{H}^p$ of $\mathcal{G}^p$:

$$\mathbf{H}^p = \tilde{\mathbf{A}}^p \cdot \mathrm{ReLU}(\tilde{\mathbf{A}}^p \bar{\mathbf{H}}^p \mathbf{W}_1^p)\mathbf{W}_2^p, \tag{5}$$

where $[\mathrm{ReLU}(\mathbf{x})]_i = \max([\mathbf{x}]_i, 0)$, $\tilde{\mathbf{A}}^p = \mathbf{D}^{p^{-\frac{1}{2}}}(\mathbf{I} + \mathbf{A}^p)\mathbf{D}^{p^{-\frac{1}{2}}}$ with $[\mathbf{D}^p]_{ii} = \sum_j [\mathbf{A}^p]_{ij}$, and $\mathbf{W}_1^p, \mathbf{W}_2^p$ are parameters to learn. Similar to (3), let $[\mathbf{s}_i^p]_j = 1$ if v_j^p appears in $\mathbf{x}_i$ and 0 otherwise. Then, we obtain $\mathbf{e}_i^p$ which encodes syntactic knowledge in $\mathcal{G}^p$ as

$$\mathbf{e}_i^p = c^p \mathbf{H}^{p\top} \mathbf{s}_i^p, \tag{6}$$

where $c^p = 1/\|\mathbf{H}^{p\top}\mathbf{s}_i^p\|_2$ is used to normalize $\mathbf{e}_i^p$ to unit norm.

4.3.3 Generic Knowledge Encoding. We already obtain the query embeddings which encode semantic knowledge from external medical KG and syntactic knowledge from POS tags. However, they do not model the semantic meaning of words in the medical search queries. Therefore, we additionally extract query embeddings by

PLMs introduced in Section 3.1. These PLMs capture generic knowledge by learning from large-scale corpus, and can also encode contextual information of each input sequence. Thus, they have become the standard workhorse for nowadays natural language processing tasks. We use BERT [9] to extract the query embedding. For each query $\mathbf{x}_i$, BERT takes "[CLS]$\mathbf{x}_i$ [SEP]" as input where [CLS]and [SEP]are special start and end tokens, and outputs embeddings for each token. One can either take the token embedding of [CLS]or the average of all word embeddings as the representation $\mathbf{e}_i^g$ for $\mathbf{x}_i$. We take the latter as $\mathbf{e}_i^g$ following the suggestion of Reimers and Gurevych [33].

4.3.4 Query Embedding. Finally, we set the query embedding $\mathbf{e}_i$ as a combination of $\mathbf{e}_i^s$ in (4) which encodes semantic knowledge, $\mathbf{e}_i^p$ in (6) which encodes syntactic knowledge, and $\mathbf{e}_i^g$ which encodes generic and contextual knowledge. Specifically, we let

$$\mathbf{e}_i = \mathbf{e}_i^s \parallel \mathbf{e}_i^p \parallel \mathbf{e}_i^g, \tag{7}$$

where $\parallel$ means concatenating vectors along the last dimension. Here, concatenation is just an example which already obtains good performance in practice. It can be replaced by a more complex aggregation function such as weighted average.

4.4 Training and Inference

The co-click query extractor has no learnable parameter. As for the query encoder, we denote it as $f_{\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\theta}$. Thus, a query $\mathbf{x}_i$ obtains its query embedding by

$$\mathbf{e}_i = f_{\boldsymbol{\theta}}(\mathbf{x}_i). \tag{8}$$

Note that (8) applies to every query in $\mathcal{S}_t$, $\{C_t^c\}$ and $\mathcal{Q}_t$.

Following Snell et al. [34], we first calculate the class prototype $\mathbf{p}_c$ of class c as

$$\mathbf{p}_c = \frac{1}{|\mathcal{S}_t^c|} \sum\nolimits_{\mathbf{x}_i \in \mathcal{S}_t^c} f_{\boldsymbol{\theta}}(\mathbf{x}_i), \tag{9}$$

where $\mathcal{S}_t^c$ is a subset of queries in $\mathcal{S}_t$ belong to class c and $|\mathcal{S}_t^c|$ denotes the number of samples in $\mathcal{S}_t^c$. Then, we obtain the class prediction for each query $\mathbf{x}_j \in \mathcal{Q}_t$. The possibility of $\mathbf{x}_j \in \mathcal{Q}_t$ over class c is estimated as

$$p(c|\mathbf{x}_j) = \frac{\exp(-d(f_{\boldsymbol{\theta}}(\mathbf{x}_j), \mathbf{p}_c))}{\sum_{c'=1}^{N} \exp(-d(f_{\boldsymbol{\theta}}(\mathbf{x}_j), \mathbf{p}_{c'}))}, \tag{10}$$

where $d(\cdot,\cdot)$ is a distance function which is set to squared Euclidean distance in this paper following [34]. The classification loss $\ell_{\text{class}}(\mathcal{Q}_t)$ on $\mathcal{Q}_t$ is calculated as

$$\ell_{\text{class}}(\mathcal{Q}_t) = \sum\nolimits_{(\mathbf{x}_j, y_j) \in \mathcal{Q}_t} - \log p(y_j|\mathbf{x}_j). \tag{11}$$

To leverage the co-click queries, we further design a regularization term based on contrastive learning. As users may click the URL by accident or find that URL cannot satisfy their needs, taking co-click queries as labeled queries can bring in noise and consequently lead to performance drop. Hence, we do not directly assign labels to those co-click queries in $\{C_t^c\}$. Instead, we take these $\{C_t^c\}$ as weak supervision and leverage them by class-aware contrastive learning. We encourage $\mathbf{x}_i \in \mathcal{S}_t^c$ and $\mathbf{c}_i \in C_t^c$ to obtain similar query embeddings by loss $\ell_{\text{weak}}(\mathcal{S}_t)$:

$$\ell_{\text{weak}}(\mathcal{S}_t) = \sum_{(\mathbf{x}_i, y_i) \in \mathcal{S}_t} \sum_{\mathbf{c}_j \in C_t^{y_i}} -\log \frac{\exp(f_{\boldsymbol{\theta}}(\mathbf{x}_i) \cdot f_{\boldsymbol{\theta}}(\mathbf{c}_j)/\eta)}{\sum_{k \in \mathcal{I}} \exp(f_{\boldsymbol{\theta}}(\mathbf{x}_i) \cdot f_{\boldsymbol{\theta}}(\mathbf{c}_k)/\eta)}, \tag{12}$$

where η is the temperature hyperparameter and $\mathcal{I}$ collectively records the indices of samples in $C_t^1 \cup \dots C_t^c \cup \dots \cup C_t^N$. Note that unlike supervised contrastive learning [19], we do not constrain the similarity between $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{S}_t^c$. This is because each class only contains K labeled exemplars, forcing them to obtain similar query embeddings can easily lead to overfit.

The complete loss for $\mathcal{T}_t$ is defined as

$$L = \ell_{\text{weak}}(\mathcal{S}_t) + \beta \ell_{\text{class}}(\mathcal{Q}_t), \tag{13}$$

where β is a hyperparameter. The episodic training procedure of MEDIC is shown in Algorithm 1.

Algorithm 1 Training procedure for MEDIC.

Input: Training data from N_{train} classes, $\mathcal{G}_m$ (medical KG schema), $\mathcal{G}_p$ (POS tag graph), N (number of classes per task), K (number of labeled exemplars per class), T (number of episodes), η and β (hyperparameters for (12) and (13));

randomly initialize model parameter $\boldsymbol{\theta}$ of MEDIC;
for $t = 1, \dots, T$ **do**
    randomly sample N classes from N_{train} intent classes;
    randomly sample K and M queries from each of the N classes to form support set $\mathcal{S}_t$ and query set $\mathcal{Q}_t$ respectively;
    extract co-click queries $\{C_t^c\}_{c=1}^N$ as (1) for queries in $\mathcal{S}_t$;
    obtain query embedding as (7) for all queries in $\mathcal{S}_t, \mathcal{Q}_t$ and associated $\{C_t^c\}_{c=1}^N$;
    predict the classes of each $\mathbf{x}_j \in \mathcal{Q}_t$ by (10);
    optimize $\boldsymbol{\theta}$ with respect to loss (13) by gradient descent;
end for
return optimized $\boldsymbol{\theta}$.

For inference, the learned MEDIC parameterized by $\boldsymbol{\theta}$ is directly applied for new tasks containing a few labeled queries from N_{test} classes that are unseen during training. For each new task $\hat{\mathcal{T}}$, a support set $\hat{\mathcal{S}}$ and a query set $\hat{\mathcal{Q}}$ are provided. We then take the following steps: (i) use the co-click query extractor to extract the co-click queries for each query in $\hat{\mathcal{S}}$; (ii) feed queries in support set $\hat{\mathcal{S}}$, associated co-click query sets $\{\hat{C}^c\}$ and query set $\hat{\mathcal{Q}}$ to the same query encoder to obtain the query embeddings; (iii) calculate class prototypes for each of the N_{test} classes; and (iv) obtain the class prediction by (10). Note that $\boldsymbol{\theta}$ is not optimized during inference. Therefore, a good performance on new tasks reveals a high generalizability of the model.

5 EXPERIMENTS

In this section, we evaluate the proposed MEDIC[4] on our real medical search query dataset (Section 2). We take five classes which contain a few samples as testing classes, including Healthcare, No Effect, Form, Drug-Action, Company. The rest classes are used for training. In addition, we create validation classes by randomly

[4]Codes are available at https://github.com/tata1661/MEDIC-SIGIR22.

Table 3: ACC (%) obtained on 5-way few-shot medical search query intent recognition tasks. The best results (according to the pairwise t-test with 95% confidence) are highlighted in bold.

Class (# testing samples)	Siamese Network	R2D2	DisSig	BERT avg	BERT CLS	CNN	LSTM	TL-GNN	Hype-GAT	NNID	MEDIC (Proposed)
1-shot											
Healthcare(48)	38.04	39.90	40.11	55.21	53.82	42.99	43.25	43.79	47.23	52.43	**63.77**
No Effect (63)	69.58	68.44	71.50	74.61	73.29	67.04	69.27	57.13	63.40	70.29	**79.44**
Form (76)	48.49	48.02	49.33	73.93	72.20	49.03	50.73	43.90	48.78	70.06	**82.25**
Drug-Action (103)	45.45	41.36	47.88	56.45	54.57	44.39	46.82	51.22	57.27	68.81	**74.30**
Company (137)	51.06	65.96	67.34	68.30	67.47	51.49	53.62	51.28	54.89	71.37	**82.98**
Macro Avg.	50.52	52.74	55.23	65.70	64.27	50.99	52.74	49.46	54.31	66.19	**75.61**
Micro Avg.	50.52	54.27	56.99	65.90	64.52	50.68	52.61	49.97	54.77	67.59	**76.57**
5-shot											
Healthcare (28)	45.03	45.68	46.97	60.34	58.57	47.83	49.29	48.17	52.44	61.26	**70.06**
No Effect (43)	72.89	73.72	74.06	74.22	73.41	72.56	72.85	62.74	68.18	74.10	**83.58**
Form (56)	62.49	60.78	64.47	76.79	76.68	62.68	61.08	55.68	59.97	76.77	**89.19**
Drug-Action (83)	50.50	51.39	52.32	59.95	59.05	51.92	50.87	57.22	60.45	73.68	**85.27**
Company (117)	57.44	68.12	74.42	72.09	71.05	56.74	57.42	58.44	60.56	78.11	**86.74**
Macro Avg.	57.67	59.94	62.45	68.68	67.75	58.35	58.30	56.45	60.32	72.78	**82.97**
Micro Avg.	57.51	61.43	64.71	69.09	68.21	57.85	57.72	57.34	60.74	74.79	**84.94**

sampling five classes from the training classes. Once the hyperparameters are chosen, we put the validation classes back and train the final model using all nine training classes. All results are averaged over five runs and are obtained on a 32GB NVIDIA Tesla V100 GPU.

5.1 Baselines

We compare **MEDIC** with the following methods.

- **Siamese Network** [23]: a classic few-shot learning model which uses dual networks to identify whether a pair of samples comes from the same class.
- **R2D2**[5] [3]: an efficient meta-learning model with closed-form solvers based on ridge regression.
- **DisSig**[6] [2]: a recent few-shot text classification method which leverages the distributional signatures which encodes word co-occurrence patterns to represent texts.
- **BERT** [7] [9]: a BERT fine-tuned with a linear classifier where a query is represented as the averaged word embeddings (**BERT avg**) or the CLS token embedding (**BERT CLS**).
- **CNN** [20]: a CNN designed to handle texts.
- **LSTM** [26]: a LSTM for text classification.
- **TLGNN** [8] [17]: a GNN which operates on document-level graphs where each document is modeled as a graph of word nodes and the word nodes are connected by globally learned edges. The texts are then classified by graph classification.
- **HyperGAT** [9] [10]: a hypergraph attention networks which operates on document-level hypergraphs such that higher-order interaction between words can be modeled.
- **NNID** [42]: the neural networks based intention detection model designed for medical search queries. It replaces words of original search queries by frequently occurring words of the dataset to augment labeled samples.

Among these methods, BERT, CNN, LSTM, TLGNN, HyperGAT and NNID are not invented for few-shot setting. Hence, we modify them to be trained on training tasks and fine-tuned on testing tasks. We implement Siamese Network, CNN, LSTM and NNID on our own due to the lack of public codes from the respective authors.

5.1.1 Experimental Settings. For all methods, we find hyperparameters using the validation set via grid search. For MEDIC, we set η in (12) as 0.05 and β in (13) as 0.4. We set the embedding size of all GCN layers used in MEDIC as 200. We train the model for a maximum number of 1000 episodes using Adam [21] with learning rate 10^{-3}. We early stop training if the validation loss does not decrease for 10 consecutive episodes. Dropout rate is set as 0.5.

5.1.2 Evaluation Metrics. Following [2], we evaluate the classification performance by test accuracy (ACC) computed on all query sets of testing tasks. As each test class contains varying number of queries, we report class-wise ACC, macro-averaged ACC which takes the average of class-wise ACCs, and micro-averaged ACC which is averaged over all testing samples.

[5] https://github.com/bertinetto/r2d2
[6] https://github.com/YujiaBao/Distributional-Signatures
[7] https://huggingface.co/bert-base-chinese
[8] https://github.com/LindgeW/TextLevelGNN
[9] https://github.com/kaize0409/HyperGAT

5.2 Performance Comparison

Table 3 shows the results. As can be seen, MEDIC performs the best in all classes with varying sizes. The fine-tuned BERT itself carries generic knowledge which can be helpful to understand search queries. However, it still performs worse than MEDIC on this fine-grained medical search query intent recognition tasks. As for existing few-shot methods including Siamese Network, R2D2 and DisSig can perform well on benchmark text classification datasets, they cannot handle well the short, noisy queries which lack semantic and syntactic information. This is also the reason why text classification methods including CNN, LSTM, TLGNN and HyperGAT fail. NNID is designed for the medical search query intent recognition task, which also enlarges the number of labeled queries. Unlike our method, NNID replaces words of original search queries by frequently occurring words in the same context. However, this does not guarantee the resultant queries convey the same intent, which makes it infeasible to handle intent classes with a few labeled samples. This reveals the necessity of designing appropriate query augmentation strategies.

5.3 Ablation Study

We compare MEDIC with the following variants:

- **w/o KG**, **w/o POS**, **w/o BERT**: they separately remove one query embedding $\mathbf{e}_i^\tau$ from $\tau \in \{m, p, g\}$ at a time;
- **w/o ℓ_{weak}**: it removes ℓ_{weak} in (13);
- **w/ CL**: it replaces ℓ_{weak} (12) by the classic contrastive learning loss [30], which pulls two co-click queries of the same $(\mathbf{x}_i, y_i)$ to be closer and pushes apart the others;
- **w/ SCL**: it replaces ℓ_{weak} (12) by the supervised contrastive learning loss [30], which encourages query embeddings of queries in $\mathcal{S}_t^c$ and C_t^c all to be similar;
- **LabelAug**: it directly takes the co-click queries as labeled data to be added to support set.

These variants are designed to cover all components of training MEDIC without overlapping functionalities.

Figure 4 shows the results. As can be seen, BERT provides a good starting point with the encoding of generic knowledge. Upon it, the knowledge encoding of medical KG schema and POS tag graph further bring in extra semantic and syntactic knowledge into the query embeddings, which increase the performance. Using co-click queries consistently improves the performance. Recall our ℓ_{weak} only enforces similarity constraint between $(\mathbf{x}_i, y_i) \in \mathcal{S}_t^c$ and $\mathbf{c}_j \in C_t^c$. The performance gain of MEDIC with respect to w/ CL, w/ SCL and LabelAug validates that the design of ℓ_{weak} can better leverage the co-click queries. In summary, none of the design considerations of MEDIC is dispensable.

5.4 Sensitivity Analysis

Here, we examine the effect of varying η in (12) and β in (13) on the performance.

Figure 5 shows the results. As shown, η affects the macro-averaged ACC more than micro-averaged ACC and obtains the best performance at $\eta = 0.05$. The existence of ℓ increases the performance where $\beta = 0.4$ obtains the best performance. However, a large β may make the model be overly dependent on the quality of co-click queries and got slightly worse performance.

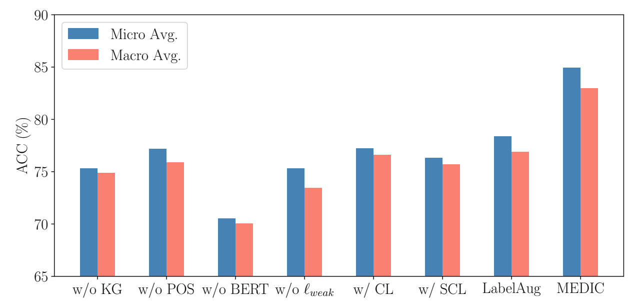


Figure 4: Ablation study on 5-way 5-shot tasks.

5.5 Effect of Varying the Number of Queries

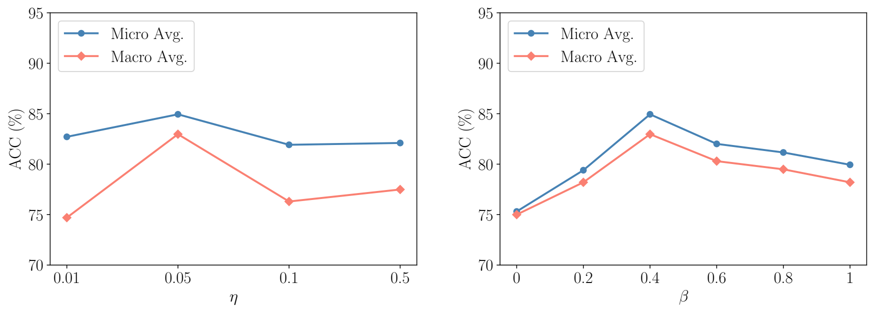
We further study the effect of varying the number of (a) labeled exemplar shots (K) in both training and testing tasks, (b) labeled exemplar shots (K) in testing tasks, and (c) co-click queries per query in (1).

The results are plotted in Figure 6. Figure 6(a) uses the same number of labeled shots K in support sets of both training and testing tasks. While Figure 6(a) trains with 5-way 5-shot tasks, but varies the number of labeled shots K in support sets of testing tasks. Despite this difference, we observe consistent performance improvement with more labeled shots. In MEDIC, recall that we only randomly chosen one co-click query from the user search query logs. Although a single co-click query per query already can obtain satisfactory result, we also try varying the number of co-click queries (Q) per query. As can be seen in Figure 6(c), the performance increases given more co-clicks queries. But the side effect is that the use of more co-click queries will increase the computational cost of (12). An interesting direction is learning to select informative co-click queries among the abundant co-click queries. However, as we tackle medical search query intent recognition tasks in a commercial search engine, heavy computational cost is not practical. How to design an efficient sampling strategy which does not incur much more computational cost than random sampling can be explored in the future.

5.6 Different Query Augmentation Strategies

Finally, we consider various query augmentation strategies to obtain the augmented data ($\{C_t^c\}$) to be used in our class-aware contrastive loss (12). In particular, we test the following strategies separately: (i) **randomly swap** and (ii) **randomly delete** [40] which randomly swap or delete words of original queries, and (iii) **rewrite** [6] which rephrases the query while maintaining a similar semantic meaning.

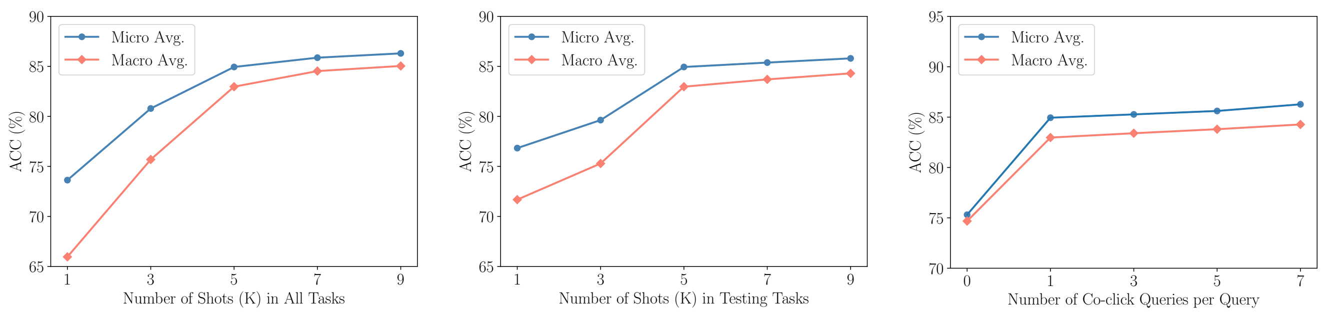
Table 5 plots the results. As can be seen, using co-click queries obtains the best performance, followed by rewrite, and then random swap and delete. To see why this happened, we provide augmented query examples for testing classes healthcare, guide and side effect in Table 4. As can be seen, the co-click queries indeed use more diverse wording and presentation style while expressing the same intent. In contrast, the other three strategies basically modify the original queries, which does not bring in much additional information. Thus, it is better to use co-click queries.



(a) η in (12).

(b) β in (13).

Figure 5: Hyperparameter sensitivity analysis of MEDIC on 5-way 5-shot tasks.



(a) ACC vs number of labled shots in all tasks.

(b) ACC vs number of labled shots in testing tasks.

(c) ACC vs number of co-click queries per query.

Figure 6: Varying the number of (a) labeled shots in training tasks, (b) labeled shots in testing tasks, and (c) co-click queries per query for 5-way tasks. For Figure 6(b), the training tasks are still 5-way 5-shot.

Table 4: Examples of queries augmented using different strategies.

	Healthcare	Guide	Side Effect
Original Query	The effect of Liuwei Dihuang teapills to males (六味地黄丸对男性功效)	Eight ciclosporin soft capsules one day (环孢素软胶囊一天8粒)	Have a slight fever after eating norfloxacin capsules (吃完诺氟沙星胶囊低烧)
Co-click Query	Is Liuwei Dihuang teapills useful for males (六味地黄丸对男性是否有用)	Six ciclosporin soft capsules one day (环孢素软胶囊一天吃6粒)	Eat norfloxacin capsules while having a fever (发烧喝诺氟沙星胶囊)
Rewrite	The benefits of Liuwei Dihuang teapills to males (六味地黄丸对男性好处)	Eat eight ciclosporin soft capsules one day (服用环孢素软胶囊一天8粒)	Have a low fever after eating norfloxacin capsules (吃完诺氟沙星胶囊低热)
Random Swap	The benefits of males to Liuwei Dihuang teapills (男性对六味地黄丸好处)	Ciclosporin soft one day eight capsules (环孢素软一天8粒胶囊)	Eat norfloxacin capsules while having a low fever (低烧 吃完诺氟沙星胶囊)
Random Delete	Liuwei Dihuang teapills to males (六味地黄丸对男性)	Eightciclosporin soft capsules (环孢素软胶囊8粒)	After eating norfloxacin capsules (吃完诺氟沙星胶囊)

Table 5: ACC(%) obtained by MEDIC using different strategies to augment queries on 5-way few-shot medical search query intent recognition tasks.

Class	Randomly Swap	Randomly Delete	Rewrite	Co-click (Proposed)
1-shot				
Healthcare	51.22	50.70	54.97	**63.77**
No Effect	65.81	65.72	68.34	**79.44**
Form	63.41	65.85	71.61	**82.25**
Drug-Action	60.54	59.09	67.36	**74.30**
Company	61.61	60.33	66.28	**82.98**
Macro Avg.	60.52	60.34	65.71	**75.61**
Micro Avg.	61.12	60.73	66.52	**76.57**
5-shot				
Healthcare	60.41	60.17	63.22	**70.06**
No Effect	73.12	70.56	76.77	**83.58**
Form	73.22	71.38	75.68	**89.19**
Drug-Action	76.09	72.02	78.25	**85.27**
Company	70.21	68.49	72.09	**86.74**
Macro Avg.	70.61	68.52	73.20	**82.97**
Micro Avg.	71.76	69.44	74.12	**84.94**

6 CONCLUSION

In this paper, we propose MEDIC to handle medical search query intent recognition task given a few labeled data. In particular, we leverage co-click queries which lead users to click the same URL as weakly supervision information to compensate for the lack of label information. As the search queries are usually noisy and lack enough semantic information and strict syntactic information, we propose a query encoder which simultaneously encodes external semantic knowledge from a medical KG, syntactic knowledge from POS tags, and generic knowledge captured from large corpus. Experimental results on a real medical search query dataset validate the effectiveness of MEDIC. We believe MEDIC can contribute to understanding the medical search query intents in an economic, effective and efficient way, and consequently promoting better online healthcare services to users all over the world. One can also use the query embeddings learned by MEDIC to measure query similarity more accurately and then perform information retrieval. In addition to retrieval, one can leverage the intent prediction results to analyze the spatial-temporal trends of a particular disease or side effect of drugs, which can help detect or monitor public health.

ACKNOWLEDGEMENTS

We sincerely thank the anonymous reviewers for their valuable comments. We also thank Zhourui Xia and Zhe Zhang who helped preprocess the raw queries. This work is supported in part by the National Key Research and Development Program of China (No. 2021ZD0110303).

REFERENCES

[1] Ioannis Antonellis, Hector Garcia-Molina, and Chi-Chao Chang. 2008. Simrank++ query rewriting through link analysis of the clickgraph. In *International World Wide Web Conference*. 1177–1178.
[2] Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot text classification with distributional signatures. In *International Conference on Learning Representations*.
[3] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. 2018. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136* (2018).
[4] Andrei Broder. 2002. A taxonomy of web search. In *ACM SIGIR Forum*, Vol. 36. ACM New York, NY, USA, 3–10.
[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, Vol. 33. 1877–1901.
[6] Hengyi Cai, Hongshen Chen, Yonghao Song, Cheng Zhang, Xiaofang Zhao, and Dawei Yin. 2020. Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. In *Annual Meeting of the Association for Computational Linguistics*. 6334–6343.
[7] Jindong Chen, Yizhou Hu, Jingping Liu, Yanghua Xiao, and Haiyun Jiang. 2019. Deep short text classification with knowledge powered attention. In *AAAI Conference on Artificial Intelligence*. 6252–6259.
[8] Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2021. An empirical survey of data augmentation for limited data learning in NLP. *arXiv preprint arXiv:2106.07499* (2021).
[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.
[10] Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be more with less: Hypergraph attention networks for inductive text classification. In *Conference on Empirical Methods in Natural Language Processing*. 4927–4936.
[11] Yuxiao Dong, Ziniu Hu, Kuansan Wang, Yizhou Sun, and Jie Tang. 2020. Heterogeneous network representation learning. In *International Joint Conference on Artificial Intelligence*, Vol. 20. 4861–4867.
[12] Thomas Dopierre, Christophe Gravier, and Wilfried Logerais. 2021. ProtAugment: Intent detection meta-learning through unsupervised diverse paraphrasing. In *Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing*. 2454–2466.
[13] Ruiying Geng, Binhua Li, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2020. Dynamic memory induction networks for few-shot text classification. In *Annual Meeting of the Association for Computational Linguistics*. 1087–1094.
[14] Rafael Glater, Rodrygo LT Santos, and Nivio Ziviani. 2017. Intent-aware semantic query annotation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 485–494.
[15] Fred X Han, Di Niu, Haolan Chen, Weidong Guo, Shengli Yan, and Bowei Long. 2020. Meta-learning for query conceptualization at web scale. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3064–3073.
[16] Michael A Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2545–2568.
[17] Lianzhe Huang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2019. Text level graph neural network for text classification. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*. 3444–3450.
[18] Dimitri Kartsaklis, Mohammad Taher Pilehvar, and Nigel Collier. 2018. Mapping text to knowledge graph entities using multi-sense LSTMs. In *Conference on Empirical Methods in Natural Language Processing*. 1959–1970.
[19] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*. 18661–18673.
[20] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Conference on Empirical Methods in Natural Language Processing*. 1746–1751.
[21] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.
[22] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.
[23] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, Vol. 2. Lille.
[24] Qian Li, Hao Peng, Jianxin Li, Congyin Xia, Renyu Yang, Lichao Sun, Philip S Yu, and Lifang He. 2020. A survey on text classification: From shallow to deep learning. *arXiv preprint arXiv:2008.00364* (2020).

[25] Ruirui Li, Liangda Li, Xian Wu, Yunhong Zhou, and Wei Wang. 2019. Click feedback-aware query recommendation using adversarial examples. In *The Web Conference*. 2978–2984.

[26] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *International Joint Conference on Artificial Intelligence*. 2873–2879.

[27] Zaiqiao Meng, Fangyu Liu, Thomas Clark, Ehsan Shareghi, and Nigel Collier. 2021. Mixture-of-Partitions: Infusing large biomedical knowledge graphs into BERT. In *Conference on Empirical Methods in Natural Language Processing*. 4672–4681.

[28] Subhabrata Mukherjee and Ahmed Awadallah. 2020. Uncertainty-aware self-training for few-shot text classification. *Advances in Neural Information Processing Systems* 33 (2020).

[29] Margi Murphy. 2019. Dr Google will see you now: Search giant wants to cash in on your medical queries. https://www.telegraph.co.uk/technology/2019/03/10/google-sifting-one-billion-health-questions-day/.

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[31] Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. Learning from context or names? An empirical study on neural relation extraction. In *Conference on Empirical Methods in Natural Language Processing*. 3661–3672.

[32] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *International World Wide Web Conference*. 91–100.

[33] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*. 3982–3992.

[34] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*. 4080–4090.

[35] Pengfei Sun, Yawen Ouyang, Wenming Zhang, and Xin-yu Dai. 2021. MEDA: Meta-learning with data augmentation for few-shot text classification. In *International Joint Conference on Artificial Intelligence*. 3929–3935.

[36] Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. 2017. Combining knowledge with deep convolutional neural networks for short text classification. In *International Joint Conference on Artificial Intelligence*. 2915–2921.

[37] Yaqing Wang, Song Wang, Quanming Yao, and Dejing Dou. 2021. Hierarchical heterogeneous graph representation learning for short text classification. In *Conference on Empirical Methods in Natural Language Processing*. 3091–3101.

[38] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *Comput. Surveys* 53, 3 (2020), 1–34.

[39] Zhongyuan Wang, Kejun Zhao, Haixun Wang, Xiaofeng Meng, and Ji-Rong Wen. 2015. Query understanding through knowledge-based conceptualization. In *International Conference on Artificial Intelligence*. 3264–3270.

[40] Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*. 6382–6388.

[41] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *AAAI Conference on Artificial Intelligence*, Vol. 33. 7370–7377.

[42] Chenwei Zhang, Wei Fan, Nan Du, and Philip S Yu. 2016. Mining user intentions from medical queries: A neural network based heterogeneous jointly modeling approach. In *International World Wide Web Conference*. 1373–1384.

[43] Hongfei Zhang, Xia Song, Chenyan Xiong, Corby Rosset, Paul N Bennett, Nick Craswell, and Saurabh Tiwary. 2019. Generic intent representation in web search. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 65–74.

[44] Jianguo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, S Yu Philip, Richard Socher, and Caiming Xiong. 2020. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *Conference on Empirical Methods in Natural Language Processing*. 5064–5082.

[45] Yi Zhao. 2020. Baidu health accelerates: Searches exceed 200 million per day, and the number of top doctors in cooperation has increased by 87a year. https://cj.sina.com.cn/articles/view/1704103183/65928d0f02001uxse?sudaref=www.google.com.hk&display=0&retcode=0.