Medical Symptom Detection in Intelligent Pre-Consultation using Bi-directional Hard-Negative Noise Contrastive Estimation

Shiwei Zhang
Tencent Jarvis Lab
Shenzhen, China
zswvizhang@tencent.com

Jichao Sun
Tencent Jarvis Lab
Shenzhen, China
jichaosun@tencent.com

Yu Huang
Tencent Jarvis Lab
Shenzhen, China
yorkeyhuang@tencent.com

Xueqi Ding
Tencent Healthcare
Shenzhen, China
sakeyding@tencent.com

Yefeng Zheng[*]
Tencent Jarvis Lab
Shenzhen, China
yefengzheng@tencent.com

ABSTRACT

Leveraging artificial intelligence (AI) techniques in medical applications is helping our world to deal with the shortage of healthcare workers and improve the efficiency and productivity of healthcare delivery. Intelligent pre-consultation (IPC) is a relatively new application deployed on mobile terminals for collecting patient's information before a face-to-face consultation. It takes advantages of state-of-the-art machine learning techniques to assist doctors on clinical decision-making. One of key functions of IPC is to detect medical symptoms from patient queries. By extracting symptoms from patient queries, IPC is able to collect more information on patient's health status by asking symptom-related questions. All collected information will be summarized as a medical record for doctors to make clinical decision. This saves a great deal of time for both doctors and patients. Detecting symptoms from patient's query is challenging, as most patients lack medical background and often tend to use colloquial language to describe their symptoms. In this work, we formulate symptom detection as a retrieval problem and propose a bi-directional hard-negative enforced noise contrastive estimation method (Bi-hardNCE) to tackle the symptom detection problem. Bi-hardNCE has both forward contrastive estimation and backward contrastive estimation, which forces model to distinguish the true symptom from negative symptoms and meanwhile distinguish true query from negative queries. To include more informative negatives, our Bi-hardNCE adopts a hard-negative mining strategy and a false-negative eliminating strategy, which achieved a significant improvement on performance. Our proposed model outperforms commonly used retrieval models by a large margin.

CCS CONCEPTS

- **Computing methodologies → Natural language processing**;
- **Information systems → Chemical and biochemical retrieval**;
- **Applied computing → Health informatics**.

KEYWORDS

medical symptom detection, pre-consultation, noise contrastive estimation, hard negatives

ACM Reference Format:
Shiwei Zhang, Jichao Sun, Yu Huang, Xueqi Ding, and Yefeng Zheng. 2022. Medical Symptom Detection in Intelligent Pre-Consultation using Bi-directional Hard-Negative Noise Contrastive Estimation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539124

1 INTRODUCTION

Intelligent pre-consultation (IPC) system is a pre-diagnosis system for improving the efficiency and productivity of healthcare delivery. The most visible function of healthcare systems is to ensure that patients and the general public can receive medical service in time. However, according to World Health Organization (WHO) report [15], the global needs-based shortage of healthcare workers including doctors, nurses and other healthcare staffs, is projected to be above 14.5 million in 2030. In the majority of low- and lower-income countries which have the greatest burdens of preventable diseases, the shortage is even worse.

To meet population health needs and expectations, countries need to employ, train and retain adequate numbers of qualified healthcare workers. Alternatively, leveraging artificial intelligence (AI) in healthcare systems also brings a significant amount of benefit. To improve the efficiency of diagnosis, natural language processing (NLP) techniques, medical knowledge graph (KG) and other AI techniques have been adopted in pre-diagnosis system to collect patient information in advance and automatically generate a draft of medical record. This saves a great deal of time for both doctors and patients. Usually, patients need to wait plenty of time for a face-to-face consultation with doctors, while the actual time spent on consultation is relatively short. In a limited time, an efficient communication between doctors and patients is a key factor influencing health outcomes. As patients often do not have a medical background, they are likely to use informal vocabularies instead

[*] Corresponding author.

KDD '22, August 14–18, 2022, Washington, DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539124

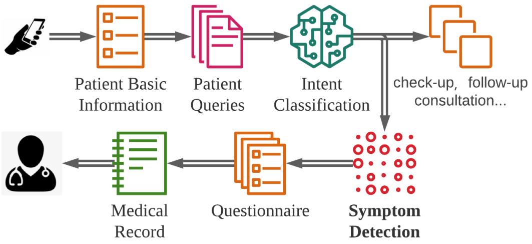


Figure 1: The overview of our intelligent pre-consultation system.

of the medical terminologies to describe their own health status. Because of this, doctors often need to interact with the patients several rounds to get a correct understanding of the patients' health status. As IPC imitates the logic of medical diagnosis, it can collect patient information in advance, which saves time for doctors from inefficient interaction with the patients. Furthermore, by modifying the generated medical report from IPC, the time for doctors to write medical records is also reduced. IPC eventually can help doctors make efficient clinical-decision and meanwhile enhance the diagnostic capability of doctors.

A pre-diagnosis system like IPC can assist doctors for pursuing a more effective and efficient diagnosis by collecting patient information on a mobile device. The framework of our IPC shown in Figure 1 has several key components, such as basic information collector, intention classifier, symptom detector, and medical record generator. After patients typing their basic information and queries, those queries will be processed by a medical intent classifier to understand patients' intentions, such as health check-up, follow-up consultation or describing symptoms. If patients' queries are about describing symptoms, the symptom detector will identify what actual symptoms are mentioned in queries. Next, IPC will ask questions related to each symptom and finally generate a medical report.

The symptom detector is the key component of IPC, as if it fails to accurately find symptoms explicitly and implicitly mentioned in patient's queries, the generated medical report will miss some of key information related to those symptoms which then likely fails to assist doctors on clinical decision-making. Detecting symptoms from patient queries is challenging as those queries tend to be informal and filled with colloquialism. For example, the first example in Table 1 is a query about describing a child has molar and otorrhea. Otorrhea is the discharge from the external part of the ear canal, while the author of this query using "water" instead of "discharge" or "pus" to describe the symptoms. In addition, patients also tend to describe symptoms by explaining what they feel and see instead of naming symptoms. For example, the patient of the fourth query in Table 1 feels that things round he/she are shaking or spinning, which actually means that the patient is experiencing vertigo. To map one of these mentioned queries into a symptom or multiple symptoms needs to overcome the challenge of lexical mismatch.

To address the aforementioned challenges in symptom detection, we first formulate it as a retrieval problem. Traditional retrieval models rely on text matching, which is hard to overcome the lexical mismatching problem. In recent years, dense retrieval models overcome this issue by matching queries and documents in a continuous dense embedding space [11], and the relevance or similarity scores are calculated simply using dot product or cosine similarity. Because of bi-encoder's efficiency, it has been widely adopted as the dense retrieval model, which has a separate encoder for query and document respectively. In terms of training a dense retrieval model, there are various methods proposed in recent years. The most popular and effective retrieval models are constructed using noise contrastive estimation (NCE) [12, 24]. One of challenges using NCE to learn a dense retrieval model is to find helpful or informative negatives for learning dense representation. Unlike unsupervised contrastive learning methods, where negatives are constructed by modifying inputs, such as rotating or cropping a given image [1], most dense retrieval models use in-batch negatives [11] and NCE as the loss. Using in-batch negatives can avoid the process of constructing negatives, however using these local negatives tends to result in a slow training convergence [22]. To overcome this issue, hard-negative mining strategies have been adopted by various recent works [6, 22, 24]. In this work, we propose a bi-directional hard-negative noise contrastive estimation (Bi-hardNCE) to identify symptoms from patient queries in IPC. The input mini-batch of our Bi-hardNCE is a batch of positive pairs (i.e., a query and the corresponding symptom contained in the query) together with its hard negatives including both hard negative queries and hard negative symptoms. When using the traditional NCE, it aims to pull positive pairs close and push a given query apart from in-batch negative symptoms. Our Bi-hardNCE not only pushes a given query apart from in-batch negative symptoms but also pushes a true symptom apart from in-batch negative queries. Additionally, hard negatives will also be included in the contrastive estimation process, which improves the robustness of retrieval models and results in a faster convergence. Following the previous work [24], the generation of hard negatives is conditioned on the retrieval model and the positive pairs. Our contributions are listed as follows:

- Based on the characteristics of medical queries in IPC, such as colloquial language, we formulate symptom detection as a retrieval problem instead of a medical concept normalization problem.
- We propose a bi-directional hard-negative enforced noise contrastive estimation for symptom detection. Our proposed model significantly outperforms baselines including popular retrieval models and the state-of-the-art NCE model using in-bath negatives.
- We evaluate our hard-negative mining strategy on baselines and find that our hard-mining strategy significantly improved the performance of all baselines, which is in consistent with the findings of the previous work [24].
- We propose a false negative eliminating (E-FN) method to tackle the false negative problem in symptom detection. With E-FN, all three hard-negative mining strategies, especially for those highly relying on top ranked negatives, have achieved significant performance gains.

Table 1: Left column shows example queries selected from IPC, while right column shows annotated symptoms by doctors.

Patient Query	Symptoms
一是小孩晚上睡觉会磨牙，声音很大。二是小孩耳朵偶有出水的情况 First, the child grinds his teeth at night and makes a loud noise. Second, there is water occasionally coming out of the child's ear.	磨牙,耳道流脓 Molar,Otorrhea
眼皮上长了小疙瘩 Pimples on eyelid.	麦粒肿 Hordeolum
感觉喉咙有酸水从胃里上来 Feel that there is acid liquid in the throat coming up from the stomach.	反酸 Sour Regurgitation
经常感觉周围的东西在晃动或者旋转，而且站不稳 Feel that things around me are shaking or spinning, and unsteady on feet.	眩晕 Vertigo
早上空腹起来能摸到肠子硬邦邦鼓起来，老是听见肚子响 When I wake up on an empty stomach in the morning, I can feel my intestines bulging hard, and always hear the sound of my stomach.	腹胀 Abdominal Distention

- Our proposed model has been deployed on a real-world IPC application. By helping IPC collect more patient health information, our model saves time for doctors from inefficient interaction with patients and eventually facilitates clinical decision-making.

2 RELATED WORK

Symptom detection is a key function of a medical pre-consultation system. Leveraging AI and machine learning techniques in medical pre-consultation to facilitate clinical decision-making is an emerging area. As many companies are exploring this new area, there are a few different designs and implementations of medical pre-consultation systems deployed in practice. However, there is not much attention on medical pre-consultation from the research community. In this section, we discuss related work including recent published papers on medical pre-consultation, medical concept detection and normalization, and noise contrastive estimation.

Medical pre-consultation or pre-diagnosis systems: A recent work [16] has developed a pre-consultation system that leverages AI techniques to build a "question-answer" system to mimic the diagnosis logic between doctors and patients. The overall structure of their system is similar to ours. Patients use mobile terminals to input their queries and symptomatic information is extracted from the query by AI techniques, and at the end of the pre-consultation a medical record will be generated. Details of AI techniques used in their system are not provided, so we cannot directly compare our models with theirs. One of core functionalities of IPC is to help patients find the right doctor. From this perspective, online pre-diagnosis system targeting on doctor recommendation [10] is close to our IPC. Another recent work [25] has proposed a neural pre-diagnosis framework for providing online medical support, which has convolutional neural networks followed by recurrent neural networks. Their framework is designed to help patients find the appropriate department, medicines, and other clinic guidance as well as pre-diagnosis suggestions.

Medical concept detection and normalization: In terms of symptom detection, it is more related to work focusing on biomedical entity normalization. Biomedical entity normalization usually has two steps, namely named entity recognition (NER) and entity normalization. It requires a model to first recognize a named medical entity in text and then map it to a unique concept ID in a biomedical ontology. Traditional methods for entity normalization highly relied on pattern matching, such as dictionary-based and rule-based approaches [5]. In recent years, Transformer-based models [21] were more widely adopted for all kinds of ranking problems including entity normalization. A recent work [2] regarded biomedical entity normalization as a retrieval problem and adopted BERT as the ranker, which achieved the stated-of-the-art performance. Symptom detection in IPC differs from biomedical entity normalization in several ways. For example, queries under symptom detection often use colloquial language and describe symptoms by explaining what a patient feels and sees. Only using recognized named entities to map symptoms is likely to miss a lot of symptomatic information. So, we formulate it as a retrieval problem instead of a biomedical entity normalization problem. As a retrieval problem, we do not need to recognize entities. Instead, we directly map a patient's query to a unique symptom or multiple symptoms in a high dimensional space.

Noise contrastive estimation (NCE): NCE is an estimation method for parameterized statistical models [7], which is widely adopted for classification and information retrieval. It learns a similarity function which can discriminate between positive examples and the noise samples (or negative examples). NCE has been widely used and achieved a great success in various applications, such as open-domain question answering (QA) [11], word embedding [13], information retrieval [9], and representation learning [20].

3 PRELIMINARIES

In this section, we discuss the preliminaries of medical symptom detection.

Problem Formulation: Given a patient's query q and a collection of predefined symptoms, the goal of a symptom detector is to find a subset of symptoms from the predefined set relevant to the query. The collection of symptoms is represented by $D = \{s_1, s_2, ..., s_m\}$, where m is the total number of predefined symptoms. The training data includes n positive (query, symptom) pairs

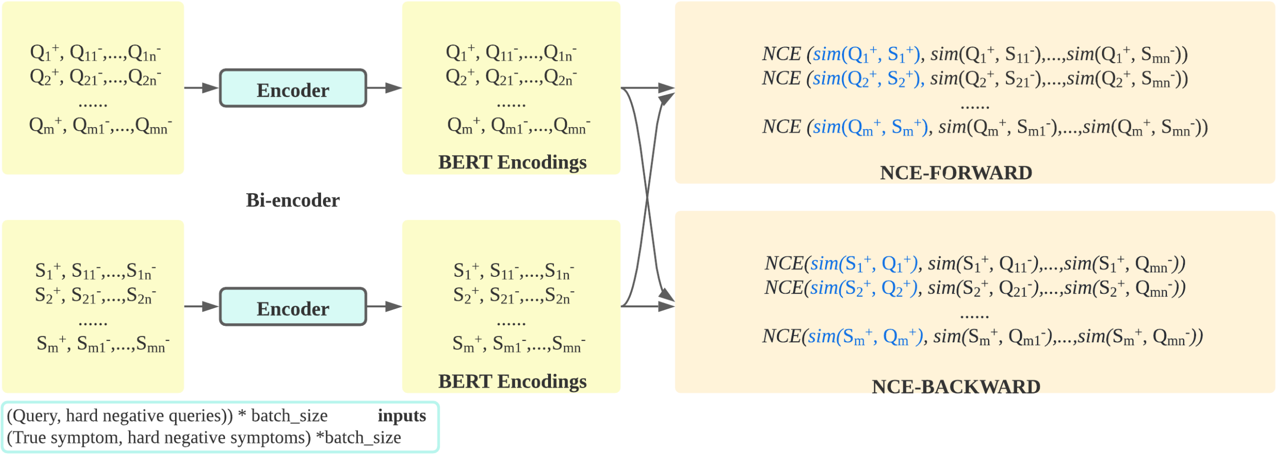

Q1+, Q11-,...,Q1n-
Q2+, Q21-,...,Q2n-
......
Qm+, Qm1-,...,Qmn-
Encoder
Bi-encoder
BERT Encodings
NCE (sim(Q1+, S1+), sim(Q1+, S11-),...,sim(Q1+, Smn-))
NCE (sim(Q2+, S2+), sim(Q2+, S21-),...,sim(Q2+, Smn-))
NCE (sim(Qm+, Sm+), sim(Qm+, Sm1-),...,sim(Qm+, Smn-))
NCE-FORWARD
S1+, S11-,...,S1n-
S2+, S21-,...,S2n-
Sm+, Sm1-,...,Smn-
Encoder
BERT Encodings
NCE(sim(S1+, Q1+), sim(S1+, Q11-),...,sim(S1+, Qmn-))
NCE(sim(S2+, Q2+), sim(S2+, Q21-),...,sim(S2+, Qmn-))
NCE(sim(Sm+, Qm+), sim(Sm+, Qm1-),...,sim(Sm+, Qmn-))
NCE-BACKWARD
(Query, hard negative queries)) * batch_size
(True symptom, hard negative symptoms) *batch_size
inputs

Figure 2: Intuitively, forward NCE aims to train the model to accurately identify the true symptom from all symptoms in a mini-batch, while backward NCE forces the model to identify the right query from all queries in a mini-batch.

$T = \{(q_1, s_1^+), (q_2, s_2^+), ...(q_n, s_n^+)\}$ [1]. We want to learn an embedding function $f : \chi \rightarrow \mathbb{S}$ that maps an observation x to a point on an embedding space $\mathbb{S}$. Given a positive pair (q_i, s_i^+), we expect that on this embedding space $\mathbb{S}$, query q_i should be close to the true symptom s_i^+ and far apart from the negative symptom s_j^-. Therefore, under the general NCE framework, the training objective is then defined by minimizing:

$$Loss_{forward} = -\log \frac{e^{sim(q_i, s_i^+)/\tau}}{e^{sim(q_i, s_i^+)/\tau} + \sum_{j=1}^{N} (e^{sim(q_i, s_j^-)/\tau})}, \qquad (1)$$

where $sim(.,.)$ is the similarity function, e.g., dot product and cosine similarity, N is the number of negative pairs, and τ is a temperature hyperparameter. To further improve the expressive ability of embedding function f, we will add backward NCE and meanwhile extend the training framework to include hard negative queries and hard negative symptoms, which will be described in Section 4. To avoid confusion with backward NCE, we denote this loss as $Loss_{forward}$.

Bi-encoder: Also known as dual encoder or Siamese-BERT [18], bi-encoder is a two-tower architecture. The model of each tower is a BERT alike encoder that encodes query and document (symptom in our case) separately. The output of each encoder is a fixed-sized sentence embedding, usually a dense vector pooled from the last layer of the encoder. The two most popular pooling strategies are directly using the vector of the special token [CLS] or computing the mean of all output vectors at the last layer. A key property of bi-encoder is that it does not require the encoding of both query and document generated at inference time as documents can be encoded offline. This makes bi-encoder an efficient retrieval model widely adopted in online applications. Hence, we adopt bi-encoder as our embedding function f.

In-batch Negatives: Traditionally, given a positive pair or a negative pair, we often choose to minimize cross-entropy loss when training retrieval models. If there are no negative pairs available, it is common to randomly select irrelevant documents from the document pool as the negative instances. Alternatively, in-batch negatives can leverage training instances in a mini-batch to avoid manually constructing negative pairs [11]. Assuming that we have N positive pairs of question q and document d in a mini-batch, for each positive pair the documents from the rest pairs [2] are used as irrelevant documents. Suppose that Q_{emb} is the embedding matrix $(N \times M)$ of all queries in a mini-batch, where M is the dimension of encoded vectors; D_{emb} is the embedding matrix $(N \times M)$ of all documents in a mini-batch. $Scores = Q_{emb} \cdot D_{emb}^T$ is the matrix $(N \times N)$ of similarity scores. Each row of this score matrix corresponds to each query paired with it's true symptom and symptoms from other queries in this mini-batch, and hence the score of positive pairs are on the diagonal of this matrix. The loss of each instance in this mini-batch will be calculated separately using NCE loss (cf. Equation (1)). Using in-batch negatives alleviates the burden of manually constructing negative pairs by taking advantages of training examples in a mini-batch, which ends up with a supervised contrastive learning. However, in-batch negatives have been found to have the drawbacks of slow training convergence as the local negatives tend to be non-informative [22].

Hard Negatives: Hard negatives are often included in training to make the problem harder so as to get more robust feature representations. It has been found that hard negative mining is always helpful for improving performance of all neural models [24]. However, to smartly select negatives as the hard ones is a challenging task as the hardness of negatives is nontrivial to define. A recent work [19] has given two guiding principles: (1) a hard negative has to be a true negative whose label has to be different

[1] Queries containing multiple symptoms are split into multiple positive pairs.

[2] Queries with multiple symptoms are processed specially to avoid introducing false negatives.

from the anchor; (2) the helpful hard negatives should be ones close to the anchor in an embedding space. Several hard-negative mining strategies have been adopted in recent work of document retrieval, such as choosing negative samples with highest BM25 scores [12], selecting top K predictions from neural retrieval models [12] or top K that are ranked above the true label [6].

4 METHODOLOGY

We propose a bi-directional hard-negative noise contrastive estimation (Bi-hardNCE) to identify medical symptoms from online patient's queries[3]. The computation graph of our Bi-hardNCE is shown in Figure 2. The architecture of the model is still a bi-encoder. However, to include both hard negative queries and hard negative symptoms in NCE, we extend the in-batch negatives training framework to bi-directional framework. In the following sections, we first describe how we include backward in-batch negatives. Next, we introduce our hard-negative mining strategies and how hard negatives are mixed with in-batch negatives. Finally, we discuss false negatives in symptom detection and introduce our strategy to control hardness of negatives in order to reduce the negative impacts brought by false negatives.

4.1 Bi-directional Noise Contrastive Estimation (Bi-NCE)

Besides forward in-batch negatives shown in Equation (1), Bi-NCE also includes backward in-batch negatives:

$$Loss_{backward} = -\log \frac{e^{sim(q_i,s_i^+)/\tau}}{e^{sim(q_i,s_i^+)/\tau} + \sum_{j=1}^{N}(e^{sim(q_j^-,s_i)/\tau})}, \quad (2)$$

where $e^{sim(q_i,s_i^+)/\tau}$ is the distance between the query and the corresponding symptom contained in the query, same as the forward NCE loss, but $\sum_{j=1}^{N}(e^{sim(q_j^-,s_i)/\tau})$ is the sum of distances between the true symptom and the rest queries in a mini-batch. The encoding of a symptom is expected to be closer to its query rather than other queries in a mini-batch. Intuitively, for forward in-batch negatives, it aims to train the model to accurately seek the true symptom from all symptoms of a mini-batch for a given query. For backward in-batch negatives, the model is pushed to find the right query from all queries of a mini-batch for a given symptom. Finally, we minimize a combined NCE loss:

$$Loss = Loss_{forward} + Loss_{backward}. \quad (3)$$

4.2 Bi-directional Hard-Negative Noise Contrastive Estimation (Bi-hardNCE)

As previous works found that in-batch negatives are likely to be uninformative, and hard negatives chosen from the global are necessary for faster learning convergence [22] and better performance [24], we also add hard negatives in our framework.

We adopt the hard-negative mining strategy proposed by a recent work [24], denoted as HD-Sampling, which chooses to generate hard negatives from the model at the beginning of each epoch. Hard negatives of HD-Sampling follow a conditional distribution, namely conditioned on the positive pairs and the model from previous epoch. Specifically, it samples negatives from incorrect predictions that are difficult to distinguish from the positives, with probabilities proportional to $e^{sim(q_i,s_i)}$. We apply HD-Sampling to generate hard negatives for both query and symptom, and add them into Bi-NCE which ends up with our Bi-hardNCE:

$$Loss_{forward} = -\log \frac{e^{sim(q_i,s_i^+)/\tau}}{e^{sim(q_i,s_i^+)/\tau} + \sum_{j=1}^{N+N^{hard}}(e^{sim(q_i,s_j^-)/\tau})}, \quad (4)$$

$$Loss_{backward} = -\log \frac{e^{sim(q_i,s_i^+)/\tau}}{e^{sim(q_i,s_i^+)/\tau} + \sum_{j=1}^{N+N^{hard}}(e^{sim(q_j^-,s_i)/\tau})}, \quad (5)$$

where N^{hard} is the number of hard negatives added into NCE.

4.3 Eliminate False Negatives (E-FN)

As mentioned earlier, in the principle of constructing hard negatives, a hard negative has to be a true negative. However, the training data often contains false negatives (or unlabeled positives), especially in document retrieval. For example, RocketQA [17] found that 70% of top-retrieved passages that were not labeled as positives in the original MSMARCO dataset [14], are actually positives. In medical symptom detection, it is also possible that true symptoms of queries are not completely annotated because of the huge quantity. In addition, it is possible that some of medical symptoms are hypernym or hyponym of others. For a given positive pair, it is likely to contain hypernym or hyponym of the true symptom in predicted hard negatives. For the sake of an effective symptom detector, it is necessary to remove false negatives from the sampled negatives.

In previous work [17], a separate model (i.e., a cross-encoder) was trained to rank negatives which are negative predictions of a retrieval model. Only negatives with high confidence scores being positives are selected as false negatives. Different from their work, we still use predicted probabilities (i.e., similarity scores) by our retrieval model in consistent with hard-negative mining strategy. We propose to tune a similarity threshold on validation data. Negatives with probabilities above a certain threshold β are considered to be false negatives, which then will be excluded from hard-negative sampling process. So the prediction of false negatives is also conditioned on the model of previous epoch. Specifically, we tune threshold β on similarity scores of validation data to assure that the precision of retrieved positives is no less than α. For example, if the target precision α is set to be 0.8, it means that there should be at least 80% of those predicted positives as true positives. We denote our false-negative eliminating method as E-FN.

4.4 Training Procedure

As shown in Figure 3, the training procedure of our framework has the following steps:

- Step 1: Initializing Bi-hardNCE with a pre-trained Chinese BERT.
- Step 2: Using HD-Sampling without E-FN to sample global negatives to enhance the original training data which only contains positive pairs. Because the model is not fine-tuned on training data yet, the outputted scores are not similarity

[3] Our code is publicly available at https://github.com/zswvivi/bihardnce.

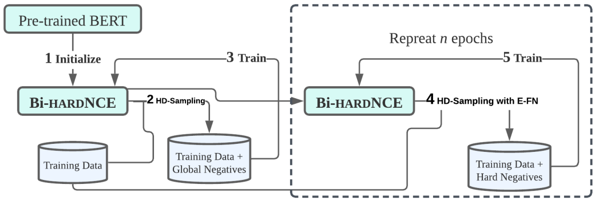


Figure 3: Training procedure: there are 5 steps, while only step 4 and step 5 will be repeated n epochs.

scores. For instance, these sampled negatives are considered as global negatives instead of hard negatives. In practice, the performance of Bi-hardNCE with sampled global negatives is about 6% higher in RECALL@5 than Bi-hardNCE without sampled global negatives at first epoch.

- Step 3: Training Bi-hardNCE using global-negative enforced training data, which ends up with a trained model M_{θ}^{prev}.
- Step 4: Generating hard negatives from M_{θ}^{prev} using both HD-Sampling and E-FN for training instances.
- Step 5: Initializing Bi-hardNCE with weights of M_{θ}^{prev}, and training it with hard-negative enforced training data.
- Repeat Step 4 to Step 5 n epochs.

5 EXPERIMENTS

In this section, we first introduce how we collect data for evaluating our framework, and then describe experimental setup. After that, we discuss experiment results.

5.1 Data

To evaluate our model, we collect a Chinese corpus from our IPC system containing about 40k online medical queries (all queries are de-identified). Then each query is annotated by doctors to check whether the query has medical symptoms or looking for other medical needs, such as follow-up consultation or check-up. If the query contains medical symptoms, doctors annotate it with a list of symptoms appeared in the query. However, before we start annotation, it is necessary to predefine a symptom list. One reason is that the actual number of medical symptoms is huge, while only part of them is needed in the pre-consultation part. In addition, doctors from different hospitals or departments tend to use synonyms or different vocabularies of a symptom which are duplicates. To avoid duplicated symptoms, we predefine a standard symptom list which will be used for annotating queries. Specifically, we ask doctors from each department to provide a list of key symptoms for pre-consultation purpose, and all of them are carefully checked and merged as one single list. The final list contains 429 medical symptoms. Next, given a query, doctors annotate it with symptoms that are in the predefined list. Finally, we end up with about 22k medical queries that have medical symptoms. The average number of symptoms in queries is 1.67 and the average length of queries is 31.17. We randomly select 2k queries as the test data, and the rest are used for training and validation. To fit the bi-encoder structure (e.g., encoding one symptom at a time), for query in training and validation sets containing more than one symptom, it is paired with each of symptoms as separate positive pairs. We have 33k positive pairs in training data and 3.6k positive pairs and 3.6k negative pairs in validation data. The negative pairs of validation data are generated by randomly selecting an irrelevant symptom for a given query. In terms of negative pairs in training data, we use various negative constructing strategies for baselines and our model, which will be described in Section 5.2. The test data remains unchanged. Given a test query, a model is required to return top 5 related symptoms. We evaluate all symptom detectors on test data using normalized discounted cumulative gain (NDCG@5) and RECALL@5 as evaluation metrics.

5.2 Experimental Setup

In this section, we first describe baseline approaches and then introduce details and implementations of our approach.

Baseline Approaches:

- Sent-BERT [18]: This is a Siamese-BERT with a softmax classifier on top of the concatenated features of two BERT branches. In terms of constructing negative pairs, we randomly select a symptom from the symptom list for a given query. The loss function is cross-entropy.
- Cross-encoder [4]: Unlike Bi-encoder, Cross-encoder has only one BERT branch, and the input is the concatenation of query and symptom with a special token [SEP] to separate them. It has a softmax classifier on top of the feature vector of the special token [CLS]. The negative pairs of training data are same as Sent-BERT. The loss function is cross-entropy.
- Triplet [3] : The input is a triplet of (query, true symptom, negative symptom), where the negative symptom is randomly selected from the symptom list. The triplet loss minimizes the distance between query and true symptom and meanwhile maximizes the distance between query and negative symptom: $max(0, sim(q, s^{+}) - sim(q, s^{-}) + margin)$.
- Contrastive [8]: Given a pair of query and symptom, it first calculates the distances using cosine similarity between the query and the symptom. If it is a positive pair, it maximizes the distance, otherwise minimizes. The negative pairs of training data are same as Sent-BERT.
- NCE-forward [11]: This uses in-batch negatives and NCE loss shown in Equation (1).

Our model Bi-hardNCE: This is an extended version of NCE-forward, which has both forward and backward NCE losses, details described in Section 4.1. In terms of hard negatives, we add both hard negative queries and hard negative symptoms. The number of hard negatives is tuned on validation data, which is 10 and 3 for query and symptom, respectively. When adopting E-FN, the initialization of precision threshold α is 0.8, which will be slightly increased after each epoch as the model is getting more competent and confident.

All of baselines and our model use a pre-trained Chinese BERT-base [4] as the initializing weights. The training is performed on a Tesla V100 with 32 GB memory. To fit into the GPU's memory, the sequence length is set to be 64 and the batch size is 32 for models with hard negatives, and 64 for the rest. The number of training epochs is 8 for all models.

Table 2: The performance (%) of baselines and our Bi-hardNCE on symptom detection.

	NDCG@5	*RECALL*@5
Sent-BERT [18]	63.25	67.32
Cross-encoder [4]	76.01	78.23
Triplet [3]	73.89	77.59
Contrastive [8]	65.14	67.65
NCE-forward [11]	78.27	82.68
Bi-hardNCE (Ours)	**83.72**	**88.26**

5.3 Experiment Results

In Table 2, we compare the performance of our Bi-hardNCE with baselines. Overall, our Bi-hardNCE significantly outperforms all of baselines. Bi-hardNCE outperforms the best baseline NCE-forward by 5.45% in NDCG@5 and 5.58% in RECALL@5. The performance of NCE-forward using in-batch negatives is better than Sent-BERT, Cross-encoder, Contrastive and Triplet which use randomly generated negatives. This indicates that in-batch negatives are more beneficial than randomly generated negatives for training symptom detectors. When comparing models only using randomly generated negatives, the performance of Cross-encoder is better than bi-encoder based structures, Sent-BERT,Triplet, and Contrastive. This is mainly because of the advantages of cross-attention mechanism, which learns token-level dependencies among tokens of both query and symptom. In addition, Triplet has better performance than both Contrastive and Sent-BERT, which indicates that having both positives and negatives when learning similarity function brings significant benefits. Unlike Contrastive which maximizes the similarity of positive pairs (query, true symptom) but minimizes the similarity of negative pairs (query, randomly selected symptom), Triplet maximizes the distance between (query, true symptom) and (query, randomly selected symptom). Triplet conducts a direct comparison between the positive and the negative, which is intuitively instructive to learn a similarity function.

In Figure 4, we can find that the performance of our Bi-hardNCE is consistently better than all of baselines from the first epoch to the last epoch, which is in consistent with the findings of the previous work: hard negatives selected from the global lead to faster learning convergence [22] and better performance [24].

5.3.1 ***Performance of Different Hard-Negative Mining Strategies.*** To compare different hard-negative mining strategies on symptom detection, the following two methods are adopted:

- TopK [17]: Top k symptoms with highest similarity scores predicted by the model from previous epoch, are selected as hard negatives.
- HigherThanTrue [6]: It selects symptoms that are ranked higher than the true symptom as hard negatives.

In addition, we also add E-FN on each of them to evaluate whether eliminating false negatives improves the performance.

In Table 3, we compare different hard-negative mining strategies and the impacts of E-FN. Without adding hard negatives, our Bi-hardNCE becomes a Bi-NCE. So we experiment Bi-NCE with different hard-negative mining strategies. According to Table 3, we can find that Bi-NCE with TopK has mixed results. It has improved performance using *NDCG*@5 as the evaluation metric, but worse performance using *RECALL*@5. When choosing HigherThanTrue as the hard-negative mining method, both *NDCG*@5 and *RECALL*@5 have been slightly improved. With E-FN added, the performance of both is significantly improved. This indicates that hard-negative mining strategies heavily relying on high similarity scores are more easily affected by false negatives as top ranked ones are likely to be false negatives.

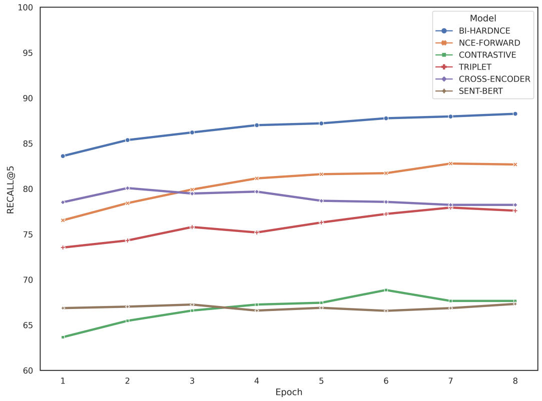


Figure 4: The performance (%) of our Bi-hardNCE and baselines at each epoch.

Table 3: Performance (%) of Bi-NCE using different hard-negative mining strategies, with or without eliminating false negatives (E-FN). It shows that E-FN is essential to TopK [17] and HigherThanTrue [6] which highly rely on top ranked negatives.

	NDCG@5	*RECALL*@5
Bi-NCE	78.39	83.31
Bi-NCE w TopK	80.15	82.88
Bi-NCE w TopK, E-FN	**83.53**	**87.30**
Bi-NCE w HigherThanTrue	80.99	84.57
Bi-NCE w HigherThanTrue, E-FN	**83.06**	**87.20**
Bi-hardNCE	**83.72**	**88.26**

5.3.2 ***Hard Negatives vs. Random Negatives.*** As we discussed before, models with random negatives tend to work worse than those with hard negatives, as shown in Table 2. So we experiment Sent-BERT, Contrastive, and Triplet with hard negatives instead of random negatives in this section, to check whether the performance of these models can be improved. We adopt the hard-negative mining strategy of our Bi-hardNCE for these three models. Specifically, we first use E-FN to eliminate false negatives and then use HD-Sampling to sample hard negatives.

In Table 4, all three models have a significant improvement on performance after including hard negatives in training. This

Table 4: Hard negatives *vs. random negatives: models with hard negatives have a significant improved performance (%) over those with random negatives.*

	NDCG@5	*RECALL*@5
Sent-BERT [18]	63.25	67.32
Sent-BERT w hard negatives	**75.08**	**79.02**
Contrastive [8]	65.14	67.65
Contrastive w hard negatives	**68.28**	**73.9**
Triplet [3]	73.89	77.59
Triplet w hard negatives	**76.11**	**79.82**

Table 5: Influences of each component of Bi-hardNCE: each time we remove a component from Bi-hardNCE and evaluate. From the result, we can find that removing the hard-negative mining strategy results in the most performance loss. In addition, the performance of NCE-forward and NCE-backward is worse than Bi-NCE.

	NDCG@5	*RECALL*@5
Bi-hardNCE	**83.72**	**88.26**
Bi-hardNCE w/o E-FN	83.63	87.69
Bi-NCE	79.39	83.31
NCE-backward	78.52	82.31
NCE-forward	78.27	82.68

is consistent with the findings of the previous work [24], which claims that hard negatives always improve the performance of models. According to Table 4, we can find that the performance of Sent-BERT with hard negatives is about 12% higher than Sent-BERT with random negatives in both *NDCG*@5 and *RECALL*@5. Additionally, both Contrastive and Triplet also have improved performance after using hard negatives, with about 6% and 2% improvement in *RECALL*@5, respectively.

5.4 Ablation Studies

In this section, we perform ablation studies over several components of our Bi-hardNCE to better understand the impacts of each component. Table 5 shows the performance of Bi-hardNCE after removing each component. The first component to be removed is E-FN. We can see that the performance is slightly dropped. Unlike TopK and HigherThanTrue, HD-Sampling is a sampling strategy which does not highly rely on top ranked predictions, so it is not strongly influenced by removing E-FN. Next, we remove hard negatives from Bi-hardNCE, which ends up with a Bi-NCE. As we discussed before, adding hard negatives is the key to improve the performance. In Table 5, it is clear that without using hard negatives the performance is dropped about 5%. The last experiment is to decompose Bi-NCE into a NCE-forward and a NCE-backward. The performance of NCE-forward and NCE-backward is roughly same, with 1% performance loss in comparison with Bi-NCE.

Table 6: The performance (%) of baselines and our Bi-hardNCE on CHIP-CDN [23].

	NDCG@5	*RECALL*@5
Sent-BERT [18]	51.18	52.03
Sent-BERT w hard negatives	63.33	64.03
Triplet [3]	60.06	61.27
Triplet w hard negatives	73.43	75.65
Contrastive [8]	48.53	49.83
Contrastive w hard negatives	69.61	72.04
NCE-forward [11]	75.04	78.35
Bi-hardNCE (Ours)	**81.61**	**85.68**

5.5 Results on Clinical Diagnosis Normalization Dataset (CHIP-CDN)

The proposed method is generic and can be applied to other short text classification tasks. To evaluate the generalization capability of our method, we experiment our model on a public data (CHIP-CDN) from the Chinese Biomedical Language Understanding Evaluation Benchmark (CBLUE) [23]. CHIP-CDN is somewhat different from the symptom detection, which is a clinical diagnosis normalization task. However, it shares similarities with the symptom detection, which is to map a non-standard diagnosis mention to one or more normalized terms. To fit into our framework, we also formulate CHIP-CDN as a retrieval task and use the same pre-processing method used by our symptom detection to construct training, validation and test data. Finally, the number of cases in each set is training (11,440), validation (2,543), and test sets (1,000). The result is shown in Table 6, we can find that (1) Bi-hardNCE outperforms all of baselines; (2) The performance of baselines with hard negatives is better than those with randomly generated negatives.

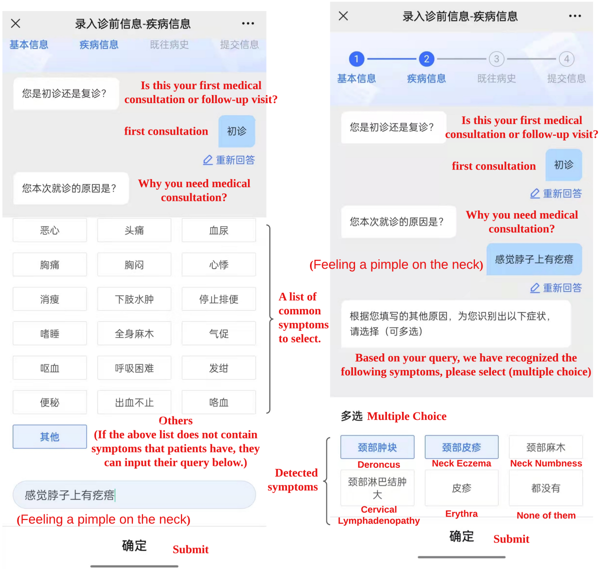
6 DEPLOYMENT

Figure 5 shows how our symptom detector applied in IPC.[4] A patient is given a short list of most common symptoms to choose from. If patients could not find their symptoms, they can choose "others" and input their query in the search bar, as shown on left side of Figure 5. The query is first processed by intent classifier, and then passed to our symptom detector if it is about describing symptoms. Our symptom detector detects symptoms from patient's query and returns a list of top 5 ranked symptoms. Then, the list will be given to the patient for a further confirmation, as shown on the right side of Figure 5. With our symptom detector, IPC has increased its capability of understanding patients' health status.

7 CONCLUSION

Symptom detection is a key component in IPC, as it extracts medical symptoms from patient queries which are the most informative patients' information in pre-consultation. Patients often use colloquial language in queries and describe their symptoms by explaining what they feel and see, which makes symptom detection

[4] https://open.tengmed.com/openAccess/ability/detail?sceneId=22&catalogId=20&serviceId=164



录入诊前信息-疾病信息

您是初诊还是复诊？ Is this your first medical consultation or follow-up visit?

初诊 first consultation

您本次就诊的原因是？ Why you need medical consultation?

感觉脖子上有疙瘩 (Feeling a pimple on the neck)

根据您填写的其他原因，为您识别出以下症状，请选择（可多选） Based on your query, we have recognized the following symptoms, please select (multiple choice)

1. Patient inputs query.

2. Patient will have a chance to confirm detected symtoms.

Figure 5: Integration of our Bi-hardNCE in IPC for symptom detection from a patient’s query.

a challenging task. To address this issue, we formulated it as a retrieval problem and proposed our Bi-hardNCE which leverages bi-directional noise contrastive estimation and hard negatives. Our model outperformed commonly used similarity models. To deal with false negatives, we proposed E-FN which removes those negatives with high confidence of being a true positive. With E-FN added, all three hard-negative mining strategies have seen performance improvement.

8 ACKNOWLEDGMENTS

We thank Shaojun Wang and Bo Chen from Tencent Electronic Health Card team for developing and marketing our IPC system. We also thank Chao Xu for helping us maintain our model in IPC.

REFERENCES

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*. PMLR, 1597–1607.

[2] Hyejin Cho, Dongha Choi, and Hyunju Lee. 2021. Re-ranking system with BERT for biomedical concept normalization. *IEEE Access* 9 (2021), 121253–121262.

[3] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*. 160–167.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[5] Jennifer D’Souza and Vincent Ng. 2015. Sieve-based entity linking for the biomedical domain. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. 297–302.

[6] Dan Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning*. 528–537.

[7] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 297–304.

[8] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2. IEEE, 1735–1742.

[9] 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*. 2333–2338.

[10] Chunhua Ju and Shuangzhu Zhang. 2021. Doctor recommendation model for pre-diagnosis online in China: integrating ontology characteristics and disease text mining. In *6th IEEE International Conference on Big Data Analytics*. IEEE, 38–43.

[11] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 6769–6781.

[12] Jing Lu, Gustavo Hernandez Abrego, Ji Ma, Jianmo Ni, and Yinfei Yang. 2021. Multi-stage training with improved negative contrast for neural passage retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 6091–6103.

[13] Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. *Advances in Neural Information Processing Systems* 26 (2013).

[14] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches NIPS*.

[15] World Health Organization et al. 2016. Health workforce requirements for universal health coverage and the sustainable development goals. (2016).

[16] Han Qian, Bin Dong, Jia-Jun Yuan, Fan Yin, Zhao Wang, Hai-Ning Wang, Han-Song Wang, Dan Tian, Wei-Hua Li, Bin Zhang, et al. 2021. Pre-consultation system based on artificial intelligence has a better diagnostic performance than the physicians in the outpatient department of pediatrics. *Frontiers in Medicine* (2021), 1907.

[17] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 5835–5847.

[18] Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, Iryna Gurevych, Nils Reimers, Iryna Gurevych, et al. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[19] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*.

[20] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints* (2018), arXiv–1807.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).

[22] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

[23] Ningyu Zhang, Mosha Chen, Zhen Bi, Xiaozhuan Liang, Lei Li, Xin Shang, Kangping Yin, Chuanqi Tan, Jian Xu, Fei Huang, et al. 2021. CBLUE: A chinese biomedical language understanding evaluation benchmark. *arXiv preprint arXiv:2106.08087* (2021).

[24] Wenzheng Zhang and Karl Stratos. 2021. Understanding hard negatives in noise contrastive estimation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 1090–1101.

[25] Xiaokang Zhou, Yue Li, and Wei Liang. 2020. CNN-RNN based intelligent recommendation for online medical pre-diagnosis support. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 18, 3 (2020), 912–921.