Contextual Fine-to-Coarse Distillation for Coarse-grained Response Selection in Open-Domain Conversations

Anonymous ACL submission

Abstract

We study the problem of coarse-grained response selection in retrieval-based dialogue systems. The problem is equally important with fine-grained response selection, but is less explored in existing literature. In this paper, we propose a **C**ontextual **F**ine-to-**C**oarse (CFC) distilled model for coarse-grained response selection in open-domain conversations. In our CFC model, dense representations of query, candidate contexts and responses is learned based on the multi-tower architecture using contextual matching, and richer knowledge learned from the one-tower architecture (fine-grained) is distilled into the multi-tower architecture (coarse-grained) to enhance the performance of the retriever. To evaluate the performance of the proposed model, we construct two new datasets based on the Reddit comments dump and Twitter corpus. Extensive experimental results on the two datasets show that the proposed method achieves huge improvement over all evaluation metrics compared with traditional baseline methods.

1 Introduction

Given utterances of a query, the retrieval-based dialogue (RBD) system aims to search for the most relevant response from a set of historical records of conversations (Higashinaka et al., 2014; Yan et al., 2016; Boussaha et al., 2019). A complete RBD system usually contain two stages: coarse-grained response selection (RS) and fine-grained response selection (Fu et al., 2020). As shown in Figure 1, in coarse-grained RS stage, the retriever identifies a much smaller list of candidates (usually dozens) from large-scale candidate database (up to millions or more), then the ranker in fine-grained RS stage selects the best response from the retrieved candidate list.

Recent studies (Whang et al., 2020; Xu et al., 2020, 2021; Whang et al., 2021) pay more attention on fine-grained RS and various complex models

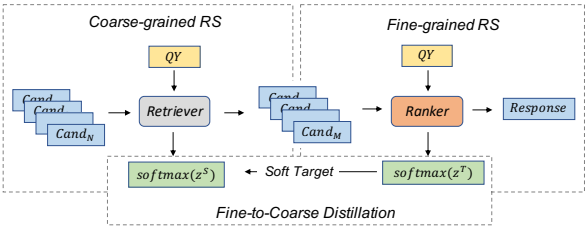


Figure 1: A common structure of retrieval-based dialogue system, where coarse-grained RS provides a much smaller ($M \ll N$) candidate set for fine-grained RS. *QY* and *Cand* are the abbreviations of *query* and *candidate* respectively.

are proposed to compute the similarities between the query and candidates for response selection. Although promising improvements have been reported, the performance of fine-grained stage is inevitably limited by the quality of the candidate list constructed. Therefore, a high-quality coarse-grained RS module is crucial, which is less explored in existing literature (Lan et al., 2020).

In this paper, we focus on the task of coarse-grained response selection, i.e., dialogue response retrieval. There are two major challenges. First, different from general text matching tasks such as ad-hoc retrieval (Hui et al., 2018) or question answering (QA) retrieval (Karpukhin et al., 2020), keywords overlapping between context and response in dialogue are potentially rare, such as when a topic transition (Sevegnani et al., 2021) occurs in response. This makes it difficult to directly match the query with candidate responses. Second, compared with fine-grained RS, coarse-grained RS deals with much larger number of candidates. Therefore, it is impractical to apply complex matching model that jointly process query and response for the similarity computation like in fine-grained RS, due to the retrieval latency (traverse millions of candidates online). Instead, the efficient BM25 system (Robertson and Zaragoza, 2009) based on sparse representations is the mainstream algorithm in coarse-

grained text matching.

To mitigate the above mentioned two problems, we propose a **C**ontextual **F**ine-to-**C**oarse (CFC) distilled model for coarse-grained RS. Instead of matching query with response directly, we propose a novel task of query-to-context matching in coarse-grained retrieval, i.e. *contextual matching*. Given a query, it is matched with candidate contexts to find most similar ones, and the corresponding responses are returned as the retrieved result. In this case, the potential richer keywords in the contexts can be utilized. To take the advantage of complex model and keep the computation cost acceptable, we distillate the knowledge learned from fine-grained RS into coarse-grained RS while maintaining the original architecture.

For the evaluation, there is no existing dataset that can be used to evaluate our model in the setting of contextual matching, because it needs to match context with context during training, while positive pairs of context-context is not naturally available like context-response pairs. Therefore, we construct two datasets based on Reddit comment dump and Twitter corpus. Extensive experimental results show that our proposed model greatly improve the retrieval recall rate and the perplexity and relevance of the retrieved responses on both datasets.

The main contributions of this paper are threefold: 1) We explore the problem of coarse-grained RS in open domain conversations and propose a Contextual Fine-to-Coarse (CFC) distilled model; 2) We construct two new datasets based on Reddit comment dump and Twitter corpus, as a new benchmark to evaluate coarse-grained RS task; 3) We construct extensive experiments to demonstrate the effectiveness and potential of our proposed model in coarse-grained RS. Both dataset and code will be released to facilitate further research on RBD systems.

2 Method

In coarse-grain response selection, there is a fixed candidate database containing a large number of *context-response* pairs. Formally, given a *query*, i.e., a new context, the goal is to retrieve Top-K most suitable *responses* for the *query* from the candidate database.

We propose a contextual fine-to-coarse distillation framework for the task of coarse-grain RS. First, we formulate the problem as a task of **contextual matching**, i.e., match query with context instead response; Second, we utilize a **multi-tower architecture** to deal with the similarity computation of query and candidates in contextual matching; Third, we utilize **knowledge distillation** to leverage the deep interaction between query and response learned in one-tower architecture.

2.1 Contextual Matching

An intuitive idea of coarse-grain RS is to treat all responses as candidate documents and directly use query to retrieve them, while this non-contextual approach results in a quite low retrieval recall rate (Lan et al., 2020). Inspired by recent studies of context-to-context matching in fine-grained RS (Fu et al., 2020), we propose contextual matching in coarse-grain RS, which is to match the query with candidate contexts, and return the responses corresponding to the most similar contexts. We consider three ways of contextual matching.

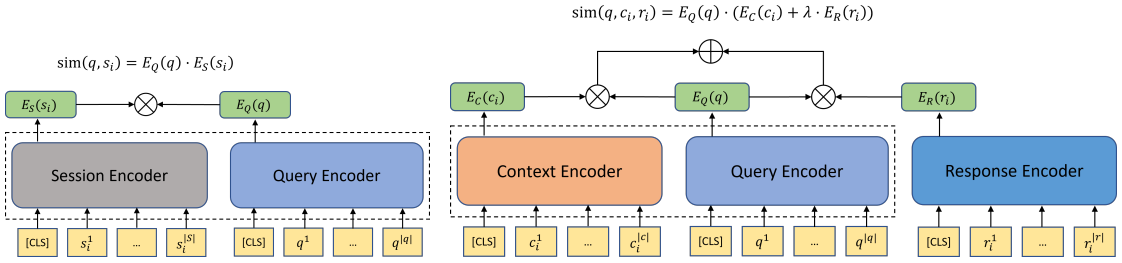
Query-Context (QC) In QC matching, we treat contexts instead of responses as candidate documents. At run-time, we calculate the similarities between query and candidate contexts, and the responses corresponding to the Top-K most similar contexts are returned as the retrieved results. The motivation of using QC matching is similar contexts may also share similar responses.

Query-Session (QS) A **session** represents the concatenated text of context and corresponding response (Fu et al., 2020), which we think more informative than context alone. In QS matching, we treat sessions as candidate documents and return the responses in Top-K most similar sessions as the retrieved results.

Decoupled Query-Session (DQS) Apart from QS matching, we also consider a decoupled way to match query with candidate sessions. In DQS matching, we treat contexts and responses as independent candidate documents. Similarities between query and contexts, query and responses are first calculated independently, then the query-session similarity can be obtained by the weighted sum. QS and DQS matching are actually two different ways to calculate query-session similarity.

2.2 Multi-Tower Architecture

For the retriever to search large-scale candidates with low latency, neural-based retrievers are usually designed as (or limited to) *multi-tower* architecture (Figure 2). In multi-tower models, the



(a) Two-tower model based on QS matching (b) Three-tower model based on DQS matching

Figure 2: Multi-tower architecture with independent encoders, the hidden representation of the [CLS] token of each sequence is passed through a linear layer followed by a hyperbolic tangent (Tanh) activation function to get the dense representations (embeddings) of the entire sentence.

query and the candidates are independently mapped to a common vector space by different encoders, where similarity can be calculated. After training, the embeddings of large-scale candidates can be **pre-calculated offline**, and only the embedding of query needs to be calculated online. In this way, fast sublinear-time approximation methods such as approximate nearest neighbor search (Shrivastava and Li, 2014) can be utilized to search for Top-K vectors that are most similar to the query, which can achieve an acceptable retrieval latency during inference.

2.2.1 Two-Tower Model

For QC and QS matching, *two-tower* architecture is adopted. Taking QS matching as an example (Figure 2(a)), the dense *session encoder* $E_S(\cdot)$ maps any candidate session to real-valued embedding vectors in a d-dimensional space, and an index is built for all the N session vectors for retrieval. At run-time, a different dense *query encoder* $E_Q(\cdot)$ maps the query to a d-dimensional vector, and retrieves k candidate sessions of which vectors are the closest to the query vector. We use the *dot product* of vectors as the similarity between query and candidate session following (Karpukhin et al., 2020).

2.2.2 Three-Tower Model

For DQS matching, dense representations of query, context and response are independently calculated, the architecture is thus designed as *three-tower* with three encoders, which is *query encoder* $E_Q(\cdot)$, *context encoder* $E_C(\cdot)$ and *response encoder* $E_R(\cdot)$ (Figure 2(b)). Similarly, context and response vectors are calculated and cached offline respectively and two indexes are built for retrieving them. The final similarity of query and session is weighted by the dot product of query-context and query-response. The weighting coefficient λ can be adjusted to determine whether it is biased to match the context or match the response[1].

2.2.3 Training Multi-Tower Model

We unify the training of the two-tower and three-tower models by formalizing them into a same *metric learning* problem (Kulis et al., 2012). The goal is to learn a matching space where similarities between positive pairs is higher than negative ones, by learning a better embedding function. We use the training of three-tower model (DQS matching) as an example. Formally, we denote the training set as $\mathcal{D} = \{q_i, \{k_i^+, k_i^-\}\}_{i=1}^N$. Each training instance contains a query q_i, a set of positive examples k_i^+ and a set of negative examples k_i^-. Among them, k_i^+ contain several positive contexts and several positive responses, similarly, k_i^- contain several negative contexts and several negative responses. We optimize the loss function as the sum of negative log likelihood of all positive pairs simultaneously:

$$\mathcal{L}(q_i) = -log\frac{\sum_{k' \in \{k_i^+\}} e^{\text{sim}(q_i, k')}}{\sum_{k' \in \{k_i^+, k_i^-\}} e^{\text{sim}(q_i, k')}} \quad (1)$$

where the similarity function is defined as:

$$\text{sim}(q_i, k') = E_Q(q_i) \cdot E(k'). \quad (2)$$

The embedding function $E(\cdot)$ of k' in Equation 2 can be $E_C(\cdot)$ or $E_R(\cdot)$, depending on the type of k'.

[1] In all experiments in this paper, λ is set to 1 to treat candidate context and response equally.

Positive and negative examples The core issue of training multi-tower models for contextual matching is to find positive pairs of query-context (or query-session). In this paper, we assume that contexts with exactly the same response are positive samples of each other, which is a cautious but reliable strategy. Formally, given a response r, if there are multiple contexts whose response is r, then we can randomly selected one context as the query q, and the other contexts are **positive contexts** of q, and r is the **positive response** of q. Negative samples of contexts and responses can be obtained from in-batch (Karpukhin et al., 2020) or random sampling from database. Similarly, positive query-session is obtained by replacing the context in positive query-context with the whole session.

2.3 Distillation from One-Tower Model

In multi-tower architecture, the query and candidates are expressed by their embeddings independently, which may cause the loss of information, and their monotonous way of interaction (inner product) further limits the capability (Lin et al., 2020). Comparing with multi-tower model, one-tower model takes both the query and the candidate as a concatenated input and allow the cross attention between query and candidate in self-attention layer. Despite fewer parameters, one-tower model have been shown to learn a more informative representations than multi-tower model, thus it is preferred in fine-grained RS (Yang and Seo, 2020). To leverage the richer expressiveness learned by the one-tower model, knowledge from one-tower model is distilled into multi-tower model to enhance the retriever.

2.3.1 Training One-Tower Model

Before distillation, we need to train teacher models based on one-tower architecture. Let's take the training of teacher model for QS matching as an example. A single encoder is trained to distinguish whether the query and the session are relevant (positive), and the form is exactly same as the next sentence prediction (NSP) task in the BERT (Devlin et al., 2018) pre-training. Formally, given a training set $\mathcal{D} = \{q_i, s_i, l_i\}_{i=1}^{N}$, where q_i is the query, s_i is the candidate session and $l_i \in \{0, 1\}$ denotes whether q_i and s_i is a positive pair. To be specific, given a query q and candidate session s, the encoder obtains the joint representation of the concatenated text of q and s, and then computes the similarity score through a linear layer, the training objective is binary cross entropy loss.

We summarize the **main difference** between one-tower and multi-tower as follows: one-tower model is more expressive, but less efficient and cannot handle large-scale candidates. The main reason is that feature-based method of calculating similarity scores rather than inner product limits the capability of offline caching. For new queries, the similarities with all candidates can only be calculated by traversal. The huge latency makes it impossible to use one-tower model in coarse-grained response retrieval. To leverage the expressiveness of one-tower model, we propose fine-to-coarse distillation, which can learn the knowledge of one-tower model while keeping the multi-tower structure unchanged, thereby improving the performance of the retriever.

2.3.2 Fine-to-Coarse Distillation

Take the two-tower **student** model (denoted as S) for QS matching as an example, suppose we have trained the corresponding one-tower **teacher** model (denoted as T). For a given query q, suppose there are a list of sessions $\{s^+, s_1^-, ..., s_n^-\}$ and the corresponding label $y = \{1, 0, ..., 0\} \in \mathcal{R}^{n+1}$, that is, one positive session and n negative sessions. We denote the similarity score vector of query-sessions computed by student model S (Equation 2) as $z^S \in \mathcal{R}^{n+1}$, then the objective of Equation 1 is equivalent to maximize the Kullback–Leibler (KL) divergence (Van Erven and Harremos, 2014) of the two distributions: $\mathrm{softmax}(z^S)$ and y, where $\mathrm{softmax}$ function turns the score vector to probability distribution.

The one-hot label y treats each negative sample equally, while the similarity between query with each negative sample is actually different. To learn more accurate labels, we further use teacher model T to calculate the similarity score vector between q and S, denoted as $z^T \in \mathcal{R}^{n+1}$. We then replace the original training objective with minimizing KL divergence of the two distributions $\mathrm{softmax}(z^S)$ and $\mathrm{softmax}(z^T)$ (Figure 1), where the temperature parameter is applied in $\mathrm{softmax}$ function to avoid saturation.

The method of fine-to-coarse distillation is to push the student model (multi-tower) to learn the predicted label of teacher model (one-tower) as a soft target instead of original one-hot label. By fitting the label predicted by the teacher model, the multi-tower model can learn a more accurate similarity score distribution from the one-tower

model while keeping the structure unchanged.

3 Datasets Construction

To evaluate the performance of the proposed model, we construct two new datasets based on the Reddit comments dump (Zhang et al., 2019) and Twitter corpus[2]. We create a training set, a multi-contexts (MC) test set and a candidate database for Reddit and Twitter respectively. For Reddit, we create an additional single-context (SC) test set. The motivation for these settings is explained in § 4.3. The size of our candidate database is one million in Twitter and ten million in Reddit respectively, which is very challenging for response retrieval. Table 1 shows the detailed statistics. We use exactly the same steps to build dataset for Reddit and Twitter, and similar datasets can also build from other large dialogue corpus in this way.

MC test set We first find out a set of responses with multiple contexts from candidate database, denoted as R. For each response r in R, we randomly select one context c from its all corresponding contexts C_r to construct a context-response (CR) pair, and put the others contexts (denoted as C_r^-) back to the database. Our MC test set consists of these CR pairs. Each response in MC test set has multiple contexts, which ensures that there exits other contexts in the database that also correspond to this response, so the retrieval recall rate can be computed to evaluate the MC test set.

SC test set We create another test set (SC) for Reddit dataset. Contrary to the MC test set, each response in SC test set has only one context, i.e., there is no context in the database that exactly corresponds to the response. Obviously, the retrieval recall rate is invalid (always zero) on SC test set. We introduce other methods to evaluate SC test set in § 4.2. The SC test set is a supplement to the MC test set which can evaluate the quality of retrieved responses given those "unique" contexts.

Candidate database To adapt to different retrieval methods, the candidate database is designed with 4 fields, namely *context*, *response*, *session*. Our candidate database consists of random context-response pairs except those in the MC and SC test sets. Besides, as mentioned above, those unselected context-response pairs (C_r^-) are deliberately merged into the database.

[2] https://github.com/Marsan-Ma-zz/chat_corpus

Datasets	Training set	Test set		Database
		MC	SC	
Reddit	300K	20K	20K	10M
Twitter	20K	2K	-	1M

Table 1: Data statistics of our new constructed datasets.

Train set The construction of training set is intuitive and similar to test set. It consists of responses and their corresponding multiple contexts. Formally, the training set can be denote as $D = \{r_i, c_{i,1}, ..., c_{i,q}\}_{i=1}^{N}$, r_i is a response and $\{c_{i,1}, ..., c_{i,q}\}$ are all contexts with response r_i, where q depends on r_i, and $q \geq 2$.

It is worth noting that there is no overlap between the contexts in the database and the contexts in the training set, which may prevent potential data leakage during training process to overestimate the evaluation metrics. The details of dataset construction are introduced in Appendix A.

4 Experiments

We conduct extensive experiments on the constructed datasets. In this section, we present experimental settings, evaluation metrics, model performance, human evaluation, etc. to demonstrate the effectiveness of the proposed models.

4.1 Compared Models

For baselines, we select BM25 (Robertson and Zaragoza, 2009) as sparse representations based method, which is widely used in real scenarios in text matching. Based on BM25 system and the two matching methods (QC and QS matching), two retrievers can be obtained, denoted as BM25-QC and BM25-QS respectively. We choose multi-tower models as dense representations based methods. They are **b**i-**e**ncoder based two-tower models for QC matching and QS matching (denoted as BE-QC and BE-QS), and **t**ri-**e**ncoder based three-tower model for DQS matching (denoted as TE-DQS). In addition, to demonstrate the advantages of contextual matching, we also report the results of query-response (QR) matching, two retrievers are build based on BM25 system and two-tower model (denoted as BM-QR and BE-QR).

There are three variants of our proposed CFC models, they are the distilled versions of BE-QC, BE-QS and TE-DQS, which are called CFC-QC, CFC-QS and CFC-DQS respectively. The distillation of each student model needs to train the

Retriever	MC Test Set								SC Test Set			
	Coverage@K				Perplexity@K		Relevance@K		Perplexity@K		Relevance@K	
	Top-1	Top-20	Top-100	Top-500	Top-1	Top-20	Top-1	Top-20	Top-1	Top-20	Top-1	Top-20
Gold	-	-	-	-	205.7		73.1		181.8		82.0	
Contextual matching												
BM25-QC	1.1	3.9	5.7	7.8	210.5	217.9	61.5	53.5	208.3	217.5	60.6	52.1
BM25-QS	0.9	3.6	5.8	8.3	207.7	214.2	80.0	73.9	200.0	208.3	81.6	74.1
BE-QC	1.3	5.3	8.1	12.3	205.4	211.5	81.3	75.8	194.4	203.2	82.9	78.3
BE-QS	1.6	5.9	11.8	20.4	200.1	206.1	85.0	80.2	190.9	199.8	85.3	80.6
TE-DQS	1.5	5.5	9.7	18.1	201.3	207.5	84.8	79.8	190.5	198.2	85.5	80.4
CFC-QC	2.9	6.5	9.1	13.0	199.5	208.9	84.9	78.6	187.5	196.3	86.2	80.8
CFC-QS	**4.2**	**7.8**	**13.1**	**21.3**	**194.8**	**203.1**	**87.8**	**82.8**	**184.3**	193.1	**88.3**	**83.4**
CFC-DQS	3.7	7.3	12.7	19.4	196.5	205.3	86.9	81.9	184.8	**192.6**	88.1	83.3
Non-contextual matching												
BM25-QR	0.2	0.7	1.3	2.4	214.2	219.2	60.3	52.9	202.8	214.5	70.4	62.7
BE-QR	0.2	0.8	1.5	2.6	207.2	213.4	72.8	67.2	198.1	206.5	78.2	71.4

Table 2: Automated evaluation metrics on Reddit test set. For MC and SC test set, we both report Perplexity@1/20 and Relevance@1/20; for SC test set, we additionally report Coverage@1/20/100/500. For Coverage@K and Relevance@K, we report the numerator of its percentage, and the larger the better; for Perplexity@K, the smaller the better.

Retriever	Coverage@K			
	Top-1	Top-20	Top-100	Top-500
BM25-QC	16.2	28.5	35.7	42.9
BM25-QS	16.3	28.3	35.1	42.8
BE-QC	19.6	36.2	46.4	56.5
BE-QS	22.1	38.9	49.7	60.2
TE-DQS	21.5	38.4	49.5	60.4
CFC-QC	24.2	39.1	48.6	58.2
CFC-QS	**28.8**	**43.7**	**52.8**	**62.6**
CFC-DQS	28.2	43.3	52.5	61.9

Table 3: Automated evaluation metrics on Twitter test set, we report Coverage@1/20/100/500 on the MC test set.

corresponding teacher model. In particular, the distillation from TE-DQS to CFC-DQS requires two teacher models, because the similarity between both query-context and query-response needs to be calculated.

We summarize the details of compared models and provide training details in Appendix B.

4.2 Evaluation Metrics

Following previous work (Xiong et al., 2020; Karpukhin et al., 2020), **Coverage@K** is used to evaluate whether Top-K retrieved candidates include the ground-truth response. It is equivalent to recall metric $R_M@K$ that often used in fine-grained RS, where N is the size of candidate database. However, Coverage@K is only suitable for evaluating the MC test set, and it is incapable for evaluating the overall retrieval quality due to the one-to-many relationship between context and response. As a supplement, we propose two automated evaluation metrics based on pre-trained models, i.e., **Perplexity@K** and **Relevance@K**. For retrieved Top-K responses, DialogGPT (Zhang et al., 2019) is used to calculate the conditional perplexity of the retrieved response given the query. DialogGPT is a language model pre-trained on 147M multi-turn dialogue from Reddit discussion thread and thus very suitable for evaluating our created Reddit dataset. Perplexity@K is the average perplexity of Top-K retrieved responses. In addition to Perplexity, we also evaluate the correlation between the query and retrieved response. We use DialogRPT (Gao et al., 2020), which is pre-trained on large-scale human feedback data with the *human-vs-rand* task that predicts how likely the response is corresponding to the given context rather than a random response. Relevance@K is the average predicted correlation degree between query and Top-K retrieved responses. Perplexity@K and Relevance@K are average metrics based on all Top-K retrieved responses, so they can reflect the overall retrieval quality.

4.3 Overall Performance

We demonstrate the main results in Table 2 and Table 3 and discuss model performance from multiple perspectives.

Dense vs. sparse It can be seen that the performance of dense retrievers far exceed that of the BM25 system, which shows rich semantic informa-

tion of PLMs and additional training can boost the performance of the retriever. For example, compared with BM25 system, the best undistilled dense retrievers (BE-QS) have a obvious improvement in three metrics. For Coverage@K, the Top-500 recall rate of BE-QS on the MC test set of Reddit and Twitter increase by 12.1% and 17.4% absolute compared with BM25-QS. For Perplexity@K, the Top-20 average perplexity of BE-QS on the MC and SC test sets of Reddit is reduced by 8.1 and 8.5 absolute compared with BM25-QS. For Relevance@K, the Top-20 average relevance of BE-QS on the MC and SC test sets on Reddit increase by 6.3% and 6.5% absolute compared with BM25-QS. Coverage@K measures the retriever's ability to retrieve gold response, while Perplexity@K and Relevance@K measure the overall retrieval quality. Our results show the consistency of the three metrics, namely, the recall rate and the overall retrieval quality have a positive correlation.

Matching method Compared with contextual matching, query-response (QR) matching has a much lower retrieval recall rate, which is also verified in (Lan et al., 2020). We think it is because that response is usually a short text of one-sentence and contains insufficient information, and there may be little keywords that overlap with the query. Therefore, it is important to consider contextual matching in the RBD system.

Compared to QC matching, QS and DQS matching should be encouraged in practice due to the additional information provided by the response. However, the BM25 system can not make good use of the information of response, as BM25-QS model does not show obvious advantages over BM25-QC on both Reddit and Twitter datasets. In contrast, dense retrieval models can effectively utilize the response. For example, BE-QS outperforms BE-QC greatly by 7.9% absolute in terms of Top-500 response retrieval recall rate in MC test set of Reddit. For QS and DQS matching, there is little difference in performance. Especially for SC test set on Reddit and MC test set on Twitter, the performance difference is minimal. One potential advantage of DQS is that it can utilize positive query-response pairs, whose number is much larger than positive query-context pairs.

Distillation benefit We future focus on the performance gain from fine-to-coarse distillation. The distilled models achieve obvious improvement in all three metrics. An obvious pattern is that the distilled models get more larger improvement with a smaller K. Take Twitter dataset as example, the Top-500 retrieval recall rate of CFC models increase by 1.5∼2.4 after distillation, while the Top-1 retrieval recall rate increased by 4.6∼6.7. On Perplexity@K and Relevance@K, our CFC models has similar performance. The significant improvement in the retrieval recall rate at small K's is especially beneficial to fine-grained response selection, because it opens up more possibility to the ranker to choose good response while seeing fewer candidates. The above results indicate that our student models benefit from learning or inheriting fine-grained knowledge from teacher models. To more clearly demonstrate the performance gains of our model after distillation, we provide the specific values of these gains in Table 7 in Appendix C.

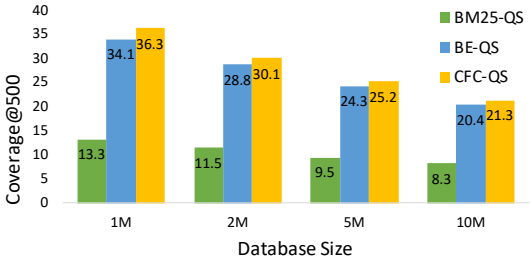


Figure 3: The Impact of database size on Coverage@500 metric of BM25-QS, BE-QS, CFC-QS.

Difference between Reddit and Twitter Since DialogGPT and DialogRPT is not pre-trained on Twitter, Perplexity@K and Relevance@K are not suitable for evaluating Twitter dataset. Therefore, we do not build SC test set for Twitter. Compared to Twitter, the Reddit dataset we use is much larger with more common multi-turn conversations, and significantly higher retrieval difficulty. The Top-500 retrieval recall rate on Twitter reach 60%, while Reddit only reached about 20%, which indicates that the coarse-grained response retrieval task in open domain conversations still has great challenges.

5 Further Analysis

5.1 Effect of Database Size

We discuss the impact of the size of candidate database on the performance of the model. For different candidate database size (from one million to ten million), we compare the Coverage@500 metric of BM25-QS, BE-QS, and CFC-QS on the

	Avg. Rank	Cohen's Kappa
CFC-QS	1.448	0.728
BE-QS	2.056	0.647
BM25-QS	2.494	0.626

Table 4: Human average rank score of BM25-QS, BE-QS and CFC-QS.

	Win	Loss	Cohen's Kappa
CFC-QS vs. BE-QS	0.747	0.253	0.634
CFC-QS vs. BM25-QS	0.816	0.184	0.672

Table 5: Human pairwise comparison of BM25-QS, BE-QS and CFC-QS.

MC test set of Reddit (Figure 3). It can be seen that Coverage@500 shows a slow downward trend as the database size increases. Increasing the size of the database will not make the model performance drop rapidly, which shows the effectiveness and robustness of our models.

5.2 Human Evaluation

To further evaluate and compare our models, we conduct a human evaluation experiment. We random select 1000 queries from the MC and SC test set (500 each) of Reddit dataset, and retrieve the Top-1 response by the BM25-QS, BE-QS and CFC-QS models respectively. Three crowd-sourcing workers are asked to score the responses. For each query, the annotator will strictly rank the retrieved responses of the three models. We report the average rank scores (between 1 and 3, the smaller the better) and the winning rate in pairwise comparison. Each two annotators have a certain number (about 200) of overlapping annotated samples. To evaluate the inter-rater reliability, the Cohen's kappa coefficient (Kraemer, 2014) is adopted.

Table 4 and Table 5 report the average ranking score of each model and pairwise comparison between models respectively. The average ranking score of CFC-QS is the highest, and CFC-QS can beat BE-QS and BM25 in most cases (74.7%∼81.6%), which indicates CFC-QS occupies a clear advantage in Top-1 retrieval. All Cohen's Kappa coefficients is between 0.6 and 0.7, indicating annotators reach moderate agreement. The results of human evaluation further verify the performance improvement brought by distillation to the model. We select several examples with human evaluation as case study and these results are presented in Appendix E.

5.3 Retrieval efficiency

We compare the retrieval latency of BM25-QS and BE-QS on the reddit MC test set, which represent the efficiency of the sparse and dense retriever respectively. We fix the batch size to 32 and retrieve top 100 most similar candidates. With the help of FAISS index, the average retrieval time of each batch by BE-QS is 581.8ms. In contrast, the average retrieval time by BM25 system using file index is 1882.6ms, about three times that of BE-QS. This indicates that the dense retriever also has an advantage in retrieval efficiency.

The relatively inferior of dense retriever is that it needs to compute the embeddings of the candidate database and establish the FAISS index, which is quite time-consuming and it takes about 9 hours for BE-QS to handle 10 million candidates with 8 GPUs, while it only takes about 10 minutes to build a BM25 index.

Since distillation does not change the structure of the retriever, it will not affect the retrieval efficiency. The cost of distillation is mainly reflected in the training of the teacher model and the extensive forward calculation in the distillation process.

6 Related Work

Fine-grained Response Selection In recent years, many works have been proposed to improve the performance of fine-grained selection module in retrieval-based chatbots (Zhang et al., 2018; Zhou et al., 2018; Tao et al., 2019; Whang et al., 2019; Yuan et al., 2019). Owing to the rapid development of pre-trained language models (PLMs) (Radford et al., 2019), recent works (Gu et al., 2020; Whang et al., 2021; Sevegnani et al., 2021) achieve the state-of-the-art (SOTA) results by utilizing PLMs such as BERT (Devlin et al., 2018) to model cross-attention and complex intersection between the context and response.

Coarse-grained Response Selection On the other hand, coarse-grained dialogue retrieval is an important but rarely explored field. Limited by efficiency, there are usually two methods for coarse-grained response selection, i.e., the sparse representations based method represented by BM25 (Robertson and Zaragoza, 2009), and the dense representations based method represented by dual-Encoder (Chidambaram et al., 2018; Humeau et al., 2019; Karpukhin et al., 2020; Lan et al., 2020; Lin et al., 2020).

Ethical Statement

In this paper, different ethical restrictions deserve discussion.

The datasets we created are derived from large dialogue corpus that publicly available on the Internet, and we strictly followed the platform's policies and rules when obtaining data from web platforms. We did not use any author-specific information in our research.

Online large dialogue corpus may includes some bias, such as political bias and social bias, and our model might have inherited some forms of these bias. In order to limit these bias as much as possible, we filter controversial articles and removed data with offensive information when possible.

References

Andrzej Białecki, Robert Muir, Grant Ingersoll, and Lucid Imagination. 2012. Apache lucene 4. In *SIGIR 2012 workshop on open source information retrieval*, page 17.

Basma El Amel Boussaha, Nicolas Hernandez, Christine Jacquin, and Emmanuel Morin. 2019. Deep retrieval-based dialogue systems: a short review. *arXiv preprint arXiv:1907.12878*.

Muthuraman Chidambaram, Yinfei Yang, Daniel Cer, Steve Yuan, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning cross-lingual sentence representations via a multi-task dual-encoder model. *arXiv preprint arXiv:1810.12836*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zhenxin Fu, Shaobo Cui, Mingyue Shang, Feng Ji, Dongyan Zhao, Haiqing Chen, and Rui Yan. 2020. Context-to-session matching: Utilizing whole session for response selection in information-seeking dialogue systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1605–1613.

Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. Dialogue response ranking training with large-scale human feedback data. *arXiv preprint arXiv:2009.06978*.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2041–2044.

Ryuichiro Higashinaka, Kenji Imamura, Toyomi Meguro, Chiaki Miyazaki, Nozomi Kobayashi, Hiroaki Sugiyama, Toru Hirano, Toshiro Makino, and Yoshihiro Matsuo. 2014. Towards an open-domain conversational system fully based on natural language processing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 928–939.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard De Melo. 2018. Co-pacrr: A context-aware neural ir model for ad-hoc retrieval. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 279–287.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Helena C Kraemer. 2014. Kappa coefficient. *Wiley StatsRef: Statistics Reference Online*, pages 1–4.

Brian Kulis et al. 2012. Metric learning: A survey. *Foundations and trends in machine learning*, 5(4):287–364.

Tian Lan, Xian-Ling Mao, Xiao-yan Gao, and He-Yan Huang. 2020. Ultra-fast, low-storage, highly effective coarse-grained selection in retrieval-based chatbot by using deep semantic hashing. *arXiv preprint arXiv:2012.09647*.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling dense representations for ranking using tightly-coupled teachers. *arXiv preprint arXiv:2010.11386*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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

Karin Sevegnani, David M Howcroft, Ioannis Konstas, and Verena Rieser. 2021. Otters: One-turn topic transitions for open-domain dialogue. *arXiv preprint arXiv:2105.13710*.

Anshumali Shrivastava and Ping Li. 2014. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). *arXiv preprint arXiv:1405.5869*.

Chongyang Tao, Wei Wu, Can Xu, Wenpeng Hu, Dongyan Zhao, and Rui Yan. 2019. Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 267–275.

Tim Van Erven and Peter Harremos. 2014. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820.

Taesun Whang, Dongyub Lee, Chanhee Lee, Kisu Yang, Dongsuk Oh, and Heuiseok Lim. 2019. Domain adaptive training bert for response selection. *arXiv preprint arXiv:1908.04812*.

Taesun Whang, Dongyub Lee, Dongsuk Oh, Chanhee Lee, Kijong Han, Dong-hun Lee, and Saebyeok Lee. 2020. Do response selection models really know what's next? utterance manipulation strategies for multi-turn response selection. *arXiv preprint arXiv:2009.04703*.

Taesun Whang, Dongyub Lee, Dongsuk Oh, Chanhee Lee, Kijong Han, Dong-hun Lee, and Saebyeok Lee. 2021. Do response selection models really know what's next? utterance manipulation strategies for multi-turn response selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14041–14049.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Ruijian Xu, Chongyang Tao, Daxin Jiang, Xueliang Zhao, Dongyan Zhao, and Rui Yan. 2020. Learning an effective context-response matching model with self-supervised tasks for retrieval-based dialogues. *arXiv preprint arXiv:2009.06265*.

Yi Xu, Hai Zhao, and Zhuosheng Zhang. 2021. Topic-aware multi-turn dialogue modeling. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*.

Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 55–64.

Sohee Yang and Minjoon Seo. 2020. Is retriever merely an approximator of reader? *arXiv preprint arXiv:2010.10999*.

Chunyuan Yuan, Wei Zhou, Mingming Li, Shangwen Lv, Fuqing Zhu, Jizhong Han, and Songlin Hu. 2019. Multi-hop selector network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 111–120.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. *arXiv preprint arXiv:1806.09102*.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1127.

A Dataset Construction Details

To filter boring and dull content and speed up the retrieval speed, we set a limit for the length of contexts and responses. We limit the context to contain at least 5 words and less than 128 words, and the response contains at least 5 words and less than 64 words. It is specially beneficial to limit the length of the response, since according to our statistics, many short responses such as "*Fair Enough*" and "*Thanks :D*" may have large number (*tens of thousands*) of different contexts.

Besides, we also limit the upper limit of the number of contexts corresponding to the response. The number of contexts of each response in the MC test set is limited to no more than 50, which is to prevent the selected responses from being a meaningless universal response. The detailed construction of the two test sets is described in Algorithm 1.

To construct the training set, we need to find out responses that corresponding multiple contexts. We use dict to implement it, where the key is the response and the value is the list of corresponding contexts. During the training of the multi-tower model, in each iteration, a batch of keys is randomly sampled from the dict. For each key (i.e., each response) in the batch, two contexts are randomly selected from the corresponding value (i.e., the list of contexts), one of which is used as the query and the other is used as a positive context, and the key is used as a positive response. The

Algorithm 1 Construction of SC & MC test set.

R: A set of unique responses.
$SC' = \varnothing$
$MC' = \varnothing$
for each $r \in R$ **do**
    $C_r = \text{FindAllContexts}(r)$ ▷ Find all contexts whose response is r.
    if $|C_r| > 1$ **then**
        $C_r^-, c = \text{Split}(C_r)$ ▷ Random pick one context c from C_r, the remaining contexts is denoted as C_r^-.
        $MC' = MC' \cup \{c, r\}$
    else
        $SC' = SC' \cup \{c \in C_r, r\}$
    end if
end for each
$MC = \text{RandomSample}(MC')$
$SC = \text{RandomSample}(SC')$
return SC, MC

other contexts and responses in the batch are all negative instances of the query.

B Model Details

Due to the different matching methods, the training of different retrievers requires slightly different input. Taking BE-QC as an example, given a query, positive and negative contexts are needed to learn the representation of query and contexts, while in BE-QS, positive and negative sessions are required. Besides, the distillation of each student model requires training corresponding teacher model, and the data of training teacher model is consistent with the student model. We summarize the input, output, and training objectives of student and teacher models in Table 6.

To implement the BM25 method, we use Elasticsearch[3], which is a powerful search engine based on Lucene library (Białecki et al., 2012). For dense retrieval methods, FAISS (Johnson et al., 2019) toolkit is used to retrieve candidate vectors. All encoders in our tower models (including one-tower, two-tower and three-tower) are initialized with *bert-base*[4], which includes 12 encoder layers, embedding size of 768 and 12 attention heads. For dense models (BE-QC, BE-QS, TE-DQS), we use the same batch size of 32 for Reddit and Twitter, and we train 30 epochs on Reddit and 10 epochs on Twitter. For all teacher models, we use the same batch size of 16, and we train 40 epochs on Reddit and 20 epochs on Twitter. For the distillation (CFC-QC, CFC-QS, CFC-DQS), we train additional 10 epochs on reddit and 5 epochs on twitter respectively, starting from the early checkpoints (20 epochs in Reddit and 5 epochs in Twitter for fair comparison) of BE-QC, BE-QS, TE-DQS. We use Adam (Kingma and Ba, 2014) optimizer with learning rate of 2e-4 and the warmup steps of 200 to optimize the parameters. We set the knowledge distillation temperature to 3 and the rate of distillation loss to 1.0. All experiments are performed on a server with 4 NVIDIA Tesla V100 32G GPUs.

[3] https://www.elastic.co/
[4] https://huggingface.co/bert-base-uncased

C Distillation Benefit

To more clearly show the performance gains of our model after distillation, we present the specific values of these gains in Table 7. Readers can compare the results in this table when reading the Distillation Benefit part in § 4.3. Positive Coverage@K and Relevance@K, and negative Perplexity@K all represent the improvement of model performance. After the distillation, the accuracy and correlation between the retrieved responses and the query increase, and the conditional perplexity decreases, indicating the huge benefits of distillation.

D Parameter Sharing

Sharing parameters in dual-encoder structure is a common practice. As shown in Figure 2, for the encoders in the dotted line, sharing parameters may be beneficial. We try parameter sharing settings on the BE-QC and TE-DQS models, respectively. We add two sets of experiments on the MC test set of Reddit, as shown in Table 8. The results show that whether or not to share parameters has little impact on Coverage@K. Therefore, we can share encoder parameters to reduce model complexity with little loss of performance.

Our guess is as follows, the sampling strategy (with replacement) create a certain probability that the query and the context are exactly the same, so the multi-tower model can learn that two identical samples are positive samples for each other, even if the parameters of the encoders are not shared.

E Case Study

As sparse representations base method, BM25 system tends to retrieve responses that overlaps with

Match	Model-ID	Architecture	Training		Inference	
			Input	Loss	Input	Output
QC	BE-QC(S)	Two-Tower	QY, POS CXT, NEG CXTs	CT	QY, CXT	DSS
	BE-QC(T)	One-Tower	QY, CXT, LABEL	CE	QY, CXT	FSS
QS	BE-QS(S)	Two-Tower	QY, POS SESS, NEG SESSs	CT	QY, SESS	DSS
	BE-QS(T)	One-Tower	QY, SESS, LABEL	CE	QY, SESS	FSS
DQS	TE-DQS(S)	Three-Tower	QY, POS CXT, NEG CXTs, POS RESP, NEG RESPs	CT	QY, CXT, RESP	DSS
	TE-DQS(T1)	One-Tower	QY, CXT, LABEL	CE	QY, CXT	FSS
	TE-DQS(T2)	One-Tower	QY, RESP, LABEL	CE	QY, RESP	FSS
QR	BE-QR	Two-Tower	QY, POS RESP, NEG RESPs	CE	QY, RESP	DSS

Abbreviation
S(Student), T(Teacher), QY(Query), CXT(Context), RESP(Response), SESS(Session), POS(Positive), NEG (Negative), CT(Contrastive), CE(Cross Entropy), DSS(Dot-product based Similarity Score), FSS(Feature based Similarity Score)

Table 6: The input, output and training objectives of tower models in this paper. For each matching method, one or two teacher models need to be trained for knowledge distillation.

Dataset	Distillation			Coverage@K				Perplexity@K		Relevance@K	
	Before		After	Top-1	Top-20	Top-100	Top-500	Top-1	Top-20	Top-1	Top-20
Reddit	BE-QC	--→	CFC-QC	+1.6	+1.2	+1.0	+0.7	-5.9	-2.6	+3.6	+2.7
	BE-QS	--→	CFC-QS	+2.6	+1.9	+1.3	+0.9	-5.3	-3.0	+2.8	+2.7
	TE-DQS	--→	CFC-DQS	+2.3	+1.8	+2.9	+1.3	-4.9	-2.1	+2.1	+2.1
Twitter	BE-QC	--→	CFC-QC	+4.6	+2.9	+2.2	+1.7	-	-	-	-
	BE-QS	--→	CFC-QS	+6.7	+4.8	+3.1	+2.4	-	-	-	-
	TE-DQS	--→	CFC-DQS	+6.7	+4.9	+3.0	+1.5	-	-	-	-

Table 7: Model performance gain after distillation on the MC test set of Reddit and Twitter dataset.

Retriever	Coverage@K			
	Top-1	Top-20	Top-100	Top-500
BE-QC	1.31	5.28	8.12	12.26
↪ share	1.29	5.26	8.12	12.26
TE-DQS	1.47	**5.52**	**9.74**	**18.12**
↪ share	**1.49**	5.51	9.73	18.11

Table 8: Impact of parameter sharing on model performance.

the context. For some complicated cases, BM25 cannot correctly retrieve those seemingly unrelated, but are the best answer in the current context.

In second case of Table 9, BM25 selects the response that contains "Spider Man 2099" in the query. But in the context of the forum, "Can I get Spider Man 2099" is actually looking for the e-book files of this comic. Compared to the comments of Spider Man 2099 given by BM25, our model retrieves "You got it PM (private message) sent!" is a harder to find, but more accurate response.

The third case is an in-game item trading query. In related forums, "keys" are used as currency. "Knife Scorched FT" and "19keys" in query respectively represent an item to be sold and its expected price. The result of BM25 covers "knife" and "key", but the meaning of the whole sentence does not match the query. On the other hand, our model selected "I only have 15keys", a standard bargaining, perfectly match the query.

There are also some examples such as case 4. Our model gives worse results than BM25. In case 4, CFC-QS retrieves a worse result, and the response retrieved by BE-QS is relatively better.

F Conclusion

In this paper, we propose a Contextual Fine-to-Coarse (CFC) distilled model. In CFC model, we adopt matching on both query-response and query-context. Considering the retrieval latency, we use multi-tower architecture to learn the **dense representations** of queries, responses and corresponding contexts. To further enhance the performance of the retriever, we distill the knowledge learned by the one-tower architecture (fine-grained) into the multi-tower architecture (coarse-grained). We construct two new datasets based on Reddit comment dump and Twitter corpus, and extensive experimental results demonstrate the effectiveness and potential of our proposed model. In the future work, we will further explore how the enhancement of coarse-grained RS can help fine-grained RS.

Number	Query	Method	Response	Rank
Case 1	My pc Isn't good enough unfortunately	Gold	How old is your computer ?	-
		CFC-QS	what are your PC specs ?	1
		BE-QS	Idk but apps aren't great on ps4 . My roku ultra is much faster for whatever reason .	2
		BM25	I' m on the edge . deals are good , but good enough to reactivate my pc ?	3
Case 2	Can I get Spider Man 2099	Gold	Good trade , thanks !	-
		CFC-QS	You got it PM sent !	1
		BE-QS	Sure , 1 by Paypal pls : xxx@hotmail.com	2
		BM25	right now , Spider man 2099 is the best written spider man .	3
Case 3	Gut Knife Scorched FT , worth 19keys	Gold	No thanks . Sorry	-
		CFC-QS	I only have 15keys .	1
		BE-QS	Add me on steam ! Nvm I added you .	2
		BM25	Nah only keys , knives are meh to me , all of'em .	3
Case 4	The email is returning failures to deliver	Gold	Should be working now .	-
		CFC-QS	THE email ? It's just email ! !	3
		BE-QS	It asks for your username I think , doesn't it ? Try just enter your username you used to register instead of the email and let me know if that works .	1
		BM25	did you get my email with the pic ?	2

Table 9: Four retrieved cases on our human evaluation set. We report Top-1 retrieved response of the three models as well as gold response. The Rank column is the ranking of the three responses given by the annotator (the lower the better).