Will this Question be Answered? Question Filtering via Answer Model Distillation for Efficient Question Answering

Siddhant Garg and **Alessandro Moschitti**
Amazon Alexa AI
{sidgarg,amosch}@amazon.com

Abstract

In this paper we propose a novel approach towards improving the efficiency of Question Answering (QA) systems by filtering out questions that will not be answered by them. This is based on an interesting new finding: the answer confidence scores of state-of-the-art QA systems can be approximated well by models solely using the input question text. This enables preemptive filtering of questions that are not answered by the system due to their answer confidence scores being lower than the system threshold. Specifically, we learn Transformer-based question models by distilling Transformer-based answering models. Our experiments on three popular QA datasets and one industrial QA benchmark demonstrate the ability of our question models to approximate the Precision/Recall curves of the target QA system well. These question models, when used as filters, can effectively trade off lower computation cost of QA systems for lower Recall, e.g., reducing computation by $\sim$60%, while only losing $\sim$3$-$4% of Recall.

1 Introduction

Question Answering (QA) technology is at the core of several commercial applications, e.g., virtual assistants such as Alexa, Google Home and Siri, serving millions of users. Optimizing the efficiency of such systems is vital to reduce their operational costs. Recently, there has been a large body of research devoted towards reducing the compute complexity of retrieval (Gallagher et al., 2019; Tan et al., 2019) and transformer-based QA models (Sanh et al., 2019; Soldaini and Moschitti, 2020).

An alternate solution for improving QA system efficiency aims to discard questions that will most probably be incorrectly answered by the system, using automatic classifiers. For example, Fader et al. (2013); Faruqui and Das (2018) aim to capture the grammaticality and well-formedness of questions. However, these methods do not take the specific answering ability of the target system into account. In practice, QA systems typically do not answer a significant portion of user questions since their answer scores could be lower than a confidence threshold (Kamath et al., 2020), tuned by the system to achieve the required Precision. For example, QA systems for medical domains exhibit a high Precision since providing incorrect/imprecise answers can have critical consequences for the end user. Based on the above rationale, discarding questions that will not be answered by the QA system presents a remarkable cost-saving opportunity. However, applying this idea may appear unrealistic from the outset since the QA system must first be executed to generate the answer confidence score.

In this paper, we take a new perspective on improving QA system efficiency by preemptively filtering out questions that will *not* be answered by the system, by means of a question filtering model. This is based on our interesting finding that the answer confidence score of a QA system can be well approximated solely using the question text.

Our empirical study is supported by several observations and intuitions. First, the final answer confidence score from a QA system (irrespective of its complex pipeline) is often generated by a Transformer-based model. This is because Transformer-based models are used for answer extraction in most research areas in QA with unstructured text, e.g., Machine Reading(MR) (Rajpurkar et al., 2016), and Answer Sentence Selection(AS2) (Garg et al., 2020). Second, more linguistically complex questions have a lower probability to be answered. Language complexity correlates with syntactic, semantic and lexical properties, which have been shown to be well captured by pre-trained language models (LMs) (Jawahar et al., 2019). Thus, the final answer extractor will be affected by said complexity, suggesting that we can predict which questions are likely to be unanswered just using their surface forms.

Third, pre-training transformer-based LMs on huge amounts of web data enables them to implicitly capture the frequency/popularity of general phrases[1], among which entities and concepts play a crucial role for answerability of questions. Thus, the contextual embedding of a question from a transformer LM is, to some extent, aware of the popularity of entities and concepts in the question, which impacts the retrieval quality of *good* answer candidates. This means that a portion of the retrieval complexity of a QA system can also be estimated just using the question. Most importantly, we only try to estimate the answer score from a QA system and not whether the answer provided by the system for a question is correct or incorrect (the latter being a much more difficult task).

Following the above intuitions, we distill the knowledge of QA models, using them as teachers, into Transformer-based models (students) that only operate on the question. Once trained, the student question model can be used to preemptively filter out questions whose answer score will not clear the system threshold, translating to a proportional reduction in the runtime cost of the system. More specifically, we propose two loss objectives for training two variants of this question filter: one with a regression head and one with a classification head. The former attempts to directly predict the continuous score provided by the QA system. The latter aims at learning to predict if a question will generate a score $>\tau$, which is the answer confidence threshold the QA system was tuned to.

We perform empirical evaluation for (i) showing the ability of our question models to estimate the QA system score; and (ii) testing the cost savings produced by our question filters, trading off with a drop in Recall. We test our models on two QA tasks with unstructured text, MR and AS2, using (a) three academic datasets: WikiQA, ASNQ, and SQuAD 1.1; (b) a large scale industrial benchmark, and (c) a variety of different transformer architectures such as BERT, RoBERTa and ELECTRA. Specifically for (i), we compare the Precision(Pr)/Recall(Re) curves of the original and the new QA system, where the latter uses the question model score to trade-off Precision for Recall. For (ii), we show the cost savings produced by our question filters, when operating the original QA system at different Precision values. The results show that: (i) The Pr/Re curves of the question models are close to those of the original system, suggesting that they can estimate the system scores well; and (ii) our question models can preemptively filter out $21.9-45.8\%$ questions while only incurring a drop in Recall of $3.2-4.9\%$. [2]

2 Related Work

Question Answering Prior efforts on QA have been broadly categorized into two fronts: tackling MR, and AS2. For the former, recently pre-trained transformer models (Devlin et al., 2019; Liu et al., 2019; Lan et al., 2020; Clark et al., 2020), etc. have achieved SOTA performance, sometimes even exceeding the human performance. Progress on this front has also seen the development of large-scale QA datasets like SQuAD (Rajpurkar et al., 2016), HotpotQA (Yang et al., 2018), NQ (Kwiatkowski et al., 2019), etc. with increasingly challenging types of questions. For the task of AS2, initial efforts embedded the question and candidates using CNNs (Severyn and Moschitti, 2015), weight aligned networks (Shen et al., 2017; Tran et al., 2018; Tay et al., 2018) and compare-aggregate architectures (Wang and Jiang, 2016; Bian et al., 2017; Yoon et al., 2019). Recent progress has stemmed from the application of transformer models for performing AS2 (Garg et al., 2020; Han et al., 2021; Lauriola and Moschitti, 2021). On the data front, small datasets like TrecQA (Wang et al., 2007) and WikiQA (Yang et al., 2015) have been supplemented with datasets such as ASNQ (Garg et al., 2020) having several million QA pairs.

Open Domain QA (ODQA) (Chen et al., 2017; Chen and Yih, 2020) systems involve a combination of a retriever and a reader (Semnani and Pandey, 2020) trained independently (Yang et al., 2017) or jointly (Yang et al., 2019). Efforts in ODQA transitioned from using knowledge bases for answering questions to using external text sources and web articles (Savenkov and Agichtein, 2016; Sun et al., 2018; Xiong et al., 2019; Lu et al., 2019). Numerous research works have proposed different techniques for improving the performance on ODQA (Min et al., 2019; Asai et al., 2019; Wang et al., 2019; Qi et al., 2019).

Filtering Ill-formed Questions Evaluating well-formedness and intelligibility of queries has been a popular research topic for QA systems. Faruqui

[1] Intended as general sequence of words, not necessarily specific to a grammatical theory, which LMs can capture well.

[2] Code can be accessed at https://github.com/alexa/wqa-question-filtering

and Das annotate the Paralex dataset (Fader et al., 2013) on the well-formedness of the questions. The majority of research efforts have been aimed at reformulating user queries to elicit the best possible answer from the QA system (Yang et al., 2014; Buck et al., 2017; Chu et al., 2019). A complementary line of work uses hate speech detection techniques (Gupta et al., 2020) to filter questions that incite hate on the basis of race, religion, etc.

Answer Verification QA systems sometimes use an answer validation component in addition to the system threshold, which analyzes the answer produced by the system and decides whether to answer or abstain. These systems often use external entity knowledge (Magnini et al., 2002; Ko et al., 2007; Gondek et al., 2012) for basing their decision to verify the correctness of the answer. Recently Wang et al. (2020) propose to add a new MR model to reflect on the predictions of the original MR model to decide whether the produced answer is correct or not. Other efforts (Rodriguez et al., 2019; Kamath et al., 2020; Jia and Xie, 2020; Zhang et al., 2021) have trained calibrators for verifying if the question should be answered or not. All these works are fundamentally different from our question filtering approach since they operate jointly on the question and generated answer, thereby requiring the entire computation to be performed by the QA system before making a decision. Our work operates only on the question text to preemptively decide whether to filter it or not. Thus the primary goal of these existing works is to improve the precision of the answering model by not answering when not confident, while our work aims to improve efficiency of the QA system and save runtime compute cost.

Query Performance Prediction Pre-retrieval query difficulty prediction has been previously explored in Information Retrieval (Carmel and Yom-Tov, 2010). Previous works (He and Ounis, 2004; Mothe and Tanguy, 2005; He et al., 2008; Zhao et al., 2008; Hauff, 2010) target $p(a|q, f)$, ground truth probability of an answer a to be correct, given a question q and a feature set f in input using simple linguistic (e.g., parse trees, polysemy value) and statistical (e.g., query term statistics, PMI) methods; while we target the QA-system score $s(a|q, f)$, the probability of an answer to be correct *as estimated* by the system. This task is more semantically driven than syntactically, and enables the use of large amounts of training data without human labels of answer correctness.

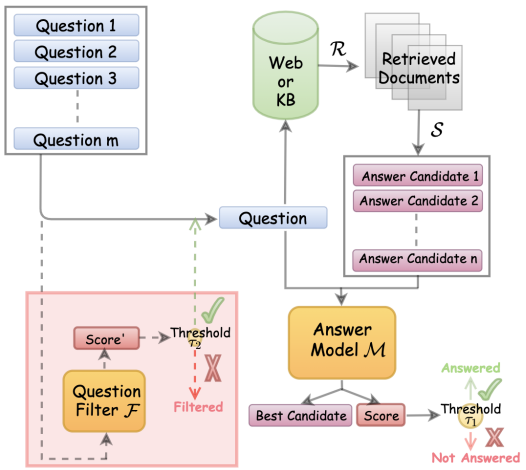


Figure 1: A real-world QA system having a retrieval ($\mathcal{R}$), candidate extraction ($\mathcal{S}$) and answering component ($\mathcal{M}$). Our proposed question filter (highlighted by the red box) preemptively removes the questions which will fail the threshold τ_1 of $\mathcal{M}$.

Efficient QA Several works on improving the efficiency of the retrieval involve using a cascade of rerankers to quickly identify good documents (Wang et al., 2011, 2016; Gallagher et al., 2019), non-metric matching functions for efficient search (Tan et al., 2019), etc. Towards reducing compute of the answer model, the following techniques have been explored: multi-stage ranking using progressively larger models (Matsubara et al., 2020), using intermediate representations for early elimination of negative candidates (Soldaini and Moschitti, 2020; Xin et al., 2020), combining separate encoding of question and answer with shallow DNNs (Chen et al., 2020), and the most popular being knowledge distillation (Hinton et al., 2015) to train smaller transformer models with low inference latencies (DistilBERT (Sanh et al., 2019), TinyBERT (Jiao et al., 2020), MobileBERT (Sun et al., 2020), etc.)

3 Preliminaries and Problem Setting

We first provide details of QA systems and explain the cost-saving opportunity space when they operate at a given Precision (or answer score threshold).

3.1 QA Systems for Unstructured Text

We consider QA systems based on unstructured text, a simple design for which works as follows (as depicted in Fig. 1): given a user question q, a search engine, $\mathcal{R}$, first retrieves a set of documents (e.g., from a web index). A text splitter, $\mathcal{S}$, applied to the documents, produces a set of passages/sentences, which are then input to an answering model $\mathcal{M}$. The latter produces the final answer. There are two

main research areas studying the design of $\mathcal{M}$:

Machine Reading (MR): $\mathcal{S}$ extracts passages $\{p_1, \ldots, p_m\}$ from the retrieved documents. $\mathcal{M}$ is a reading comprehension head, which uses $(q, \{p_1, \ldots, p_m\})$ to predict start and end position (span) for the best answer based on these passages.

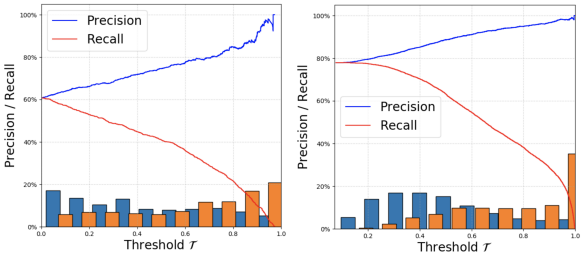
Answer Sentence Selection (AS2): $\mathcal{S}$ splits the retrieved documents into a set $\{s_1, \ldots, s_m\}$ of individual sentences. $\mathcal{M}$ performs sentence re-ranking over $(q, \{s_1, \ldots, s_m\})$, where the top ranked candidate is provided as the final answer. $\mathcal{M}$ is typically learned as a binary classifier applied to QA pairs, labelled as being correct or incorrect. AS2 models can handle scaling to large collection of sentences (more documents and candidates) more easily than MR models (the latter process passages in entirety before answering while the former can break down passages into candidates and evaluate them in parallel) thereby having lower latency at inference time.

3.2 Precision/Recall Tradeoff

$\mathcal{M}$ provides the best answer (irrespective of the answer modeling being MR or AS2) for a given question q along with a prediction score σ (DNNs typically produce a normalized probability), which is termed *MaxProb* in several works (Hendrycks and Gimpel, 2017; Kamath et al., 2020). The most popular technique to tune the Pr/Re tradeoff is to set a threshold, τ, on σ. This means that the system provides an answer for q only if $\sigma > \tau$. Henceforth, we denote $\mathcal{M}$ operating at a threshold τ by $\mathcal{M}\langle\tau\rangle$. While not calibrated perfectly (as shown by Kamath et al.), the predictions of QA models are supposed to be aligned with the ground truth such that questions that are *correctly* answered are more likely to receive higher σ than those that are *incorrectly* answered. This is an effect of the binary cross-entropy loss, $\mathcal{L}_{\text{CE}}$, typically used for training $\mathcal{M}$[3]. For example, Fig. 2 plots Pr/Re on varying threshold τ of popular MR and AS2 systems, both built using transformer models (SQuAD: BERT-Base $\mathcal{M}$, ASNQ: RoBERTa-Base $\mathcal{M}$, details in Section 5.2). The results show that increasing τ achieves a higher Pr trading it off for lower Re.

3.3 Question Filtering Opportunity Space

Real-world QA systems are always associated with a target Precision, which is typically rather high to meet the customer quality requirements[4] using the threshold τ. This means that systems will not provide an answer for a large fraction of questions (q's for which $\sigma \leq \tau$). For example from Fig. 2(b), to obtain a Precision of 90% for SQuAD 1.1, we need to set τ=0.55, resulting in not answering 35.2% of all questions. Similarly, to achieve the same Precision on ASNQ (Fig. 2(a)), we need to set τ=0.89, resulting in not answering 88.8% of all questions.



(a) ASNQ (b) SQuAD 1.1

Figure 2: Change in Pr/Re on varying threshold τ for $\mathcal{M}$. Additionally, we plot fraction of correctly/ incorrectly answered questions with the score σ of $\mathcal{M}$ in ranges $[0, 0.1], \ldots, [0.9, 1]$ on the x-axis. Frequency of correct answers increases towards the right (as $\sigma \rightarrow 1$).

It is important to note that the QA system still performs the entire computation: $\mathcal{R} \rightarrow \mathcal{S} \rightarrow \mathcal{M}$ on all questions (even the unanswered ones), to decide whether to answer or not. Thus, filtering these questions before executing the system can save the cost of running redundant computation, e.g., 35.2% or 88.8% of the cost of the two systems above (assuming the required Precision value is 90%). In the next section, we show how we build models that can produce a reliable prediction of the QA system score, only using the question text.

4 Modelling QA System Scores using Input Questions

We propose to use a distillation approach to learn a model operating on questions that can predict the confidence score of the QA system, within a certain error bound. We denote the QA system with $\Omega(\mathcal{R}, \mathcal{S}, \mathcal{M}\langle\tau\rangle)$, and the question model by $\mathcal{F}$ (as we will use it to filter out questions preemptively). Intuitively, $\mathcal{F}$ aims at learning how confident the answer model $\mathcal{M}$ is on answering a particular question when presented with a set of candidate answers from a retrieval system $(\mathcal{R}, \mathcal{S})$.

For a question q, we indicate the set of answer

[3]For MR, the sum of two cross entropy loss values is used: one for the start and one for the end of the answer. This sum is not exactly the probability of having a correct/incorrect answer, but correlates well with the probability of correctness.

[4]Even if the Recall were very low this system can still be very useful for serving a portion of customer requests, as a part of a committee of specialized QA systems each answering specific user requests.

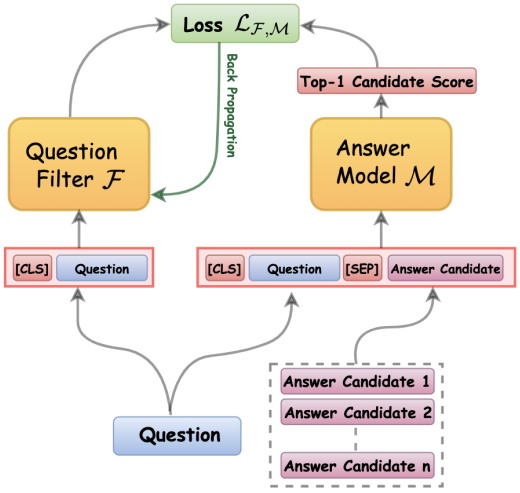


Figure 3: Distilling QA model $\mathcal{M}$ to train the question filter $\overline{\mathcal{F}_{\mathcal{M}}}$ with a regression head using the MSE $\mathcal{L}_{\mathcal{F},\mathcal{M}}$

candidate sentences/passages by $\bar{s} = \{s_1, \ldots, s_m\}$. The output from $\mathcal{M}$ for q and $\bar{s}$ corresponds to the score for the best candidate/span: $\mathcal{M}(q, \bar{s}) = \max_{s \in \bar{s}} \mathcal{M}(q, s)$. We train a filter $\overline{\mathcal{F}_{\mathcal{M}}}$ for $\mathcal{M}$ using a regression head to directly predict the score of $\mathcal{M}$ irrespective of the threshold τ using the loss:

$$\mathcal{L}_{\mathcal{F},\mathcal{M}}(q, \bar{s}) = \mathcal{L}_{\text{MSE}}\big(\mathcal{F}(q), \mathcal{M}(q, \bar{s})\big), \quad (1)$$

where $\mathcal{L}_{\text{MSE}}$ is the mean square error loss. Fig. 3 diagrammatically shows the training process of $\overline{\mathcal{F}_{\mathcal{M}}}$.

Additionally, as $\mathcal{M}$ typically operates with a threshold τ, we train a filter $\mathcal{F}_{\mathcal{M}\langle\tau\rangle}$ corresponding to a specific τ, i.e., $\mathcal{M}\langle\tau\rangle$, using the following loss:

$$\mathcal{L}_{\mathcal{F},\mathcal{M}\langle\tau\rangle}(q, \bar{s}) = \mathcal{L}_{\text{CE}}\Big(\mathcal{F}(q), \mathbb{1}\big(\mathcal{M}(q, \bar{s}) > \tau\big)\Big), \quad (2)$$

where $\mathcal{L}_{\text{CE}}$ is the cross entropy loss and $\mathbb{1}$ denotes the binary indicator function.

The novelty of our proposed approach from standard distillation techniques (Hinton et al., 2015; Sanh et al., 2019; Jiao et al., 2020) stems from the fact that, unlike the standard setting, in our case the teacher $\mathcal{M}$ and the student $\mathcal{F}$ operate on different inputs: $\mathcal{F}$ only on the questions while $\mathcal{M}$ on question-answer pairs. This makes our task much more challenging as $\mathcal{F}$ needs to approximate the probability of $\mathcal{M}$ fitting *all* answer candidates for a question. Since our $\mathcal{F}$ does not predict if an answer provided by $\mathcal{M}$ is correct/incorrect, we don't require labels of the QA dataset for training $\mathcal{F}$ ($\mathcal{F}$'s output only depends on predictions of $\mathcal{M}$). This enables large scale training of $\mathcal{F}$ without any human supervision using the predictions of a system $\Omega(\mathcal{R}, \mathcal{S}, \mathcal{M}\langle\tau\rangle)$ and a large number of questions.

To use the trained $\mathcal{F}$ for preemptively filtering out questions, we use a threshold on the score of $\mathcal{F}$. Henceforth, we refer to the threshold of $\mathcal{M}$ by τ_1 and that of $\mathcal{F}$ by τ_2. Any question q for which $\mathcal{F}(q) \leq \tau_2$, gets filtered out. Using the question filter, we define the new QA system as $\hat{\Omega}(\mathcal{F}\langle\tau_2\rangle, \mathcal{R}, \mathcal{S}, \mathcal{M}\langle\tau_1\rangle)$ where the filter $\mathcal{F}$ can be trained using Eq. 1 or 2 ($\overline{\mathcal{F}_{\mathcal{M}}}$ or $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$).

5 Experiments

First, we compare how well our models $\mathcal{F}$ can approximate the answer score of $\mathcal{M}$. Then we optimize τ_1 and τ_2 on the dev. set to precisely estimate the cost savings that we can obtain with the application of our approach to questions filtering. We also compare it with different baseline models for $\mathcal{F}$ from previous works on question filtering.

5.1 Datasets

We use three academic and one industrial datasets to validate our claims across different data domains and question answering tasks (MR and AS2).

WikiQA: An AS2 dataset (Yang et al., 2015) with questions from Bing search logs and answer candidates from Wikipedia. We use the most popular setting of training with questions having at least one positive answer candidate, and testing in the *clean* mode with questions having at least one positive and one negative answer candidate.

ASNQ: A large scale AS2 dataset (Garg et al., 2020) [5] corresponding to Natural Questions (NQ), containing over $60k$ questions and $23M$ answer candidates. Compared to WikiQA, ASNQ has more sophisticated user questions derived from Google search logs and a very high class imbalance ($\sim$ 1 correct in 400 candidate answers) thereby making it a challenging dataset for AS2. We divide the *dev* set from the release of ASNQ into two equal splits with 1336 questions each to be used for validation and testing.

SQuAD1.1: A large scale MR dataset (Rajpurkar et al., 2016) [6] containing questions asked by crowdworkers with answers derived from Wikipedia articles. Unlike the previous two datasets, SQuAD1.1 requires predicting the exact answer span to answer a question from the provided passage. We divide the *dev* set into two splits of 5266 and 5267 questions for validation and testing respectively. Pr/Re is computed based on exact answer match (EM).

AQAD: A large scale internal industrial dataset

[5] https://github.com/alexa/wqa_tanda

[6] https://rajpurkar.github.io/SQuAD-explorer/

containing *non-representative de-identified* user questions from Alexa virtual assistant. Alexa QA Dataset (AQAD) contains 1 million and 50k questions in its train and dev. sets respectively, with their top answer and confidence scores as provided by the QA system (without any human labels of correctness). Note that the top answer is selected using an answer selection model from hundreds of candidates that are retrieved from a large web-index ($\sim$ 1B web pages). For the purpose of this paper, we use a human annotated portion of AQAD (5k questions other than the train/dev. splits) as the test split for our experiments. Results on AQAD are presented relative to the baseline $\mathcal{M}\langle 0 \rangle$ due to the data being internal.

Sugawara et al. previously highlight several shortcomings of using popular MR datasets like SQuAD1.1 for evaluation, due to artifacts such as (i) 35% questions being answerable only using their first 4 tokens, (ii) 76% questions having the correct answer in the sentence with the highest unigram overlap with the question, etc. To ensure that our question filters are learning the capability of the QA system and not these artifacts, we consider datasets from industrial scenarios (where questions are real customer queries) like ASNQ, AQAD [7] and WikiQA in addition to SQuAD.

5.2 Models

For each of the three academic datasets, we use two transformer based models (12 and 24 layer) as $\mathcal{M}$: state-of-the-art RoBERTa-Base and RoBERTa-Large trained with TANDA for WikiQA [2] (Garg et al., 2020); RoBERTa-Base and RoBERTa-Large-MNLI fine-tuned on ASNQ [2] (Garg et al., 2020); and, BERT-Base and BERT-Large fine-tuned on SQuAD1.1 (Devlin et al., 2019). For AQAD, we use ELECTRA-Base trained using TANDA (Garg et al., 2020) after an initial transfer on ASNQ as $\mathcal{M}$. For the question filter $\mathcal{F}$, we use two different transformer based models (RoBERTa-Base, Large) for each of the four datasets. For WikiQA, ASNQ and SQuAD1.1, the RoBERTa-Base $\mathcal{F}$ is used for the 12-layer $\mathcal{M}$ and the RoBERTa-Large $\mathcal{F}$ is used for the 24-layer $\mathcal{M}$. For AQAD we train both the RoBERTa-Base and RoBERTa-Large $\mathcal{F}$ for the single ELECTRA-Base $\mathcal{M}$. All experimental details are presented in Appendix B, C for reproducibility.

[7] For ASNQ and IQAD, only 7.04% and 5.82% questions are answered correctly by the highest unigram overlap answer to the question respectively.

5.3 Baselines

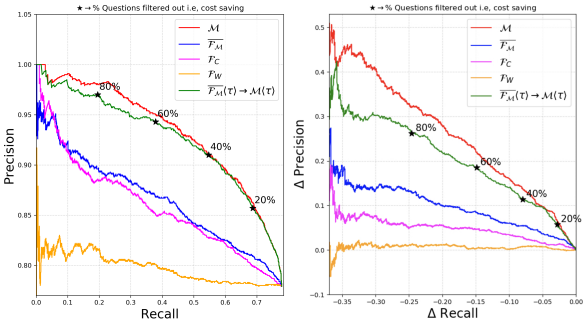
To demonstrate efficacy of our question filters, we use two question filtering baselines. The first captures well-formedness and intelligibility of questions from a human perspective. For this we train RoBERTa-Base, Large regression models on question well-formedness human annotation scores of the Paralex dataset (Faruqui and Das, 2018) [8]. We denote the resulting filter by $\mathcal{F}_{\text{W}}$. For the second baseline, we train a question classifier which predicts whether $\mathcal{M}$ will correctly answer a question. This idea has been studied in very recent contemporary works (Varshney et al., 2020; Chakravarti and Sil, 2021) but for answer verification (not for efficiency). We fine-tune RoBERTa-Base, Large for each dataset to predict whether the target $\mathcal{M}$ correctly answers the question or not. We denote this filter by $\mathcal{F}_{\text{C}}$.

We exclude comparisons with early exiting strategies (Soldaini and Moschitti, 2020; Xin et al., 2020; Liu et al., 2020) that adaptively reduce the number of transformer layers per sample and aim to improve efficiency of $\mathcal{M}$ instead of Ω. Inference batching strategy with multiple samples cannot exploit this efficiency benefit directly, thus these works report efficiency gains through abstract concepts such as FLOPs (Floating Point Operations per Second) using an inference batch-size=1, which is not practical. The efficiency gains from our approach are tangible, since filtering questions can scale down the required number of GPU-compute instances. Furthermore, ideas from these works can easily be combined with ours to add both the efficiency gains to the QA System.

5.4 Approximating Precision/Recall of $\mathcal{M}$

Firstly, we want to compare how well our question filter $\mathcal{F}$ can approximate the answer score from $\mathcal{M}$. For doing this, we plot the Pr/Re curves of $\mathcal{M}$ by varying τ_1 (i.e, $\mathcal{M}\langle\tau_1\rangle$) and that of filter $\mathcal{F}$ by varying τ_2 (i.e, $\mathcal{F}\langle\tau_2\rangle$) on the dataset test splits. We consider three options for filter $\mathcal{F}$: our regression head question filter $\overline{\mathcal{F}_{\mathcal{M}}}$ and the two baselines: $\mathcal{F}_{\text{W}}$, $\mathcal{F}_{\text{C}}$. We present graphs on SQuAD1.1 and AQAD using RoBERTa-Base $\mathcal{F}$ in Fig. 4. Note that our classification-head filter ($\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$) is trained specific to a particular τ_1 for $\mathcal{M}$, and hence it cannot be directly compared in Fig. 4 (since training $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$ for every $\tau_1 \in [0, 1]$ is not feasible).

[8] https://github.com/google-research-datasets/query-wellformedness



(a) SQuAD 1.1 (b) AQAD

Figure 4: Pr/Re curves for filters ($\overline{\mathcal{F}_{\mathcal{M}}}$, $\mathcal{F}_{\text{W}}$, $\mathcal{F}_{\text{C}}$) and answer model $\mathcal{M}$ (For AQAD we show ΔPr/ΔRe w.r.t $\mathcal{M}\langle 0\rangle$). Since test splits only contain questions with at least one correct answer, Pr=Re at τ_1=0.

The graphs show that $\overline{\mathcal{F}_{\mathcal{M}}}$ approximates the Pr/Re of $\mathcal{M}$ much better than the baseline filters $\mathcal{F}_{\text{W}}$ and $\mathcal{F}_{\text{C}}$. The gap in approximating Pr/Re of $\mathcal{M}$ between $\overline{\mathcal{F}_{\mathcal{M}}}$ and $\mathcal{F}_{\text{C}}$ indicates that learning answer scores is easier than predicting if the model's answer is correct just using the question text.

While these plots independently compare $\overline{\mathcal{F}_{\mathcal{M}}}$ and $\mathcal{M}$, in practice, $\widehat{\Omega}$ will operate $\mathcal{M}$ at a non-zero threshold τ_1 sequentially after $\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau_2\rangle$ (henceforth we denote this by $\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau_2\rangle{\rightarrow}\mathcal{M}\langle\tau_1\rangle$). To simplify visualization of the resulting system in Fig. 4, we propose to use a single common threshold τ for both $\overline{\mathcal{F}_{\mathcal{M}}}$ and $\mathcal{M}$, denoted as $\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau\rangle{\rightarrow}\mathcal{M}\langle\tau\rangle$. From Fig. 4-(a), (b), the Pr/Re curve for $\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau\rangle{\rightarrow}\mathcal{M}\langle\tau\rangle$ on varying τ approximates that of $\mathcal{M}$ very well. Using $\overline{\mathcal{F}_{\mathcal{M}}}$ however, imparts a large efficiency gain to $\widehat{\Omega}$ as shown by the four operating points that represent the % of questions filtered out by $\overline{\mathcal{F}_{\mathcal{M}}}$. For example, for AQAD, 60% of all the questions can be filtered out before running $(\mathcal{R},\mathcal{S},\mathcal{M})$ (translating to a cost saving of the same fraction) while only dropping the Recall of Ω by 3 to 4 points. Complete plots having Pr/Re curves for $\mathcal{F}_{\text{C}}\langle\tau\rangle{\rightarrow}\mathcal{M}\langle\tau\rangle$ and $\mathcal{F}_{\text{W}}\langle\tau\rangle{\rightarrow}\mathcal{M}\langle\tau\rangle$ for all four datasets are included in Appendix D.

5.5 Selecting Threshold τ_2 for $\mathcal{F}$

When adding a question filter $\mathcal{F}$ to $\Omega(\mathcal{R},\mathcal{S},\mathcal{M})$, the operating threshold τ_2 of $\mathcal{F}$ is a user-tunable parameter which can be varied per the use-case: efficiency desired at the expense of recall. This user-tunable τ_2 is a prominent advantage of our approach since one can decide what fraction of questions to filter out based on how much recall one can afford to drop. We plot the variation of the fraction of questions filtered by $\mathcal{F}$ along with the change in Pr/Re of $\widehat{\Omega}$ on varying τ_2 in Fig. 5. Specifically we consider the ASNQ and AQAD datasets, and $\mathcal{M}$ operating at τ_1=0.5. From Fig. 5(a) we can observe that for ASNQ, our filter $\overline{\mathcal{F}_{\mathcal{M}}}$ can obtain $\sim$18% filtering gains while only losing a recall of $\sim$3 points. $\overline{\mathcal{F}_{\mathcal{M}}}$ can obtain even better filtering gains on AQAD: from Fig. 5(b) $\sim$40% filtering by only losing $\sim$4 points of recall. Complete plots for all datasets can be found in Appendix E.

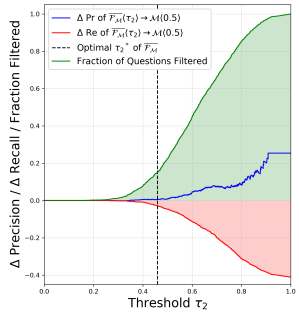
We now present one possible way to choose an operating threshold τ_2 for filter $\mathcal{F}$. For a QA system $\widehat{\Omega}(\mathcal{F}\langle\tau_2\rangle,\mathcal{R},\mathcal{S},\mathcal{M}\langle\tau_1\rangle)$, we find the threshold τ_2^* for $\mathcal{F}$ at which it best approximates the answering/abstaining choice of $\mathcal{M}\langle\tau_1\rangle$. Specifically, we use the dev. split of the datasets to find ${\tau_2}^*\in[0,1]$ such that $\mathcal{F}\langle\tau_2^*\rangle$ obtains the highest F1-score corresponding to the binary decision of answering or abstaining by $\mathcal{M}\langle\tau_1\rangle$. We present empirical results of our filters at different thresholds τ_1 of $\mathcal{M}$ in Table 1. We evaluate the % of questions filtered out by $\mathcal{F}\langle\tau_2^*\rangle$ (efficiency gains) and the resulting drop in recall of $\mathcal{F}\langle\tau_2^*\rangle{\rightarrow}\mathcal{M}\langle\tau_1\rangle$ from $\mathcal{M}\langle\tau_1\rangle$ on the test split of the dataset. For each dataset and model $\mathcal{M}$, we train one regression head filter $\overline{\mathcal{F}_{\mathcal{M}}}$ and five classification head filters $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$: one at every threshold τ_1 for $\mathcal{M}\in\{0.3,0.5,0.6,0.7,0.9\}$. For regression head $\overline{\mathcal{F}_{\mathcal{M}}}$, the optimal ${\tau_2}^*$ is calculated independently for every τ_1 of $\mathcal{M}$.

Results: From Table 1 we observe that our question filters (both with classification and regression heads) can impart filtering efficiency gains (of different proportions) while only incurring a small drop in recall. For example on ASNQ (12-layer $\mathcal{M}$, $\mathcal{F}$), $\mathcal{F}_{\mathcal{M}\langle 0.5\rangle}$ is able to filter out 17.8% of the questions while only incurring a drop in recall of 2.9%. On ASNQ (24-layer $\mathcal{M}$, $\mathcal{F}$), $\overline{\mathcal{F}_{\mathcal{M}}}$ is able to filter out 21.6% of the questions with a drop in recall of only 3.9% at τ_1=0.7. Barring some cases at higher thresholds, $\overline{\mathcal{F}_{\mathcal{M}}}$ achieves comparable filtering performance to $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$. The best filtering gains are obtained on the industrial AQAD dataset having real world noise, where for τ_1= 0.5, the 24-layer $\mathcal{F}_{\mathcal{M}\langle 0.5\rangle}$ can filter out 45.8% of all questions only incurring a drop in recall of 4.9%.

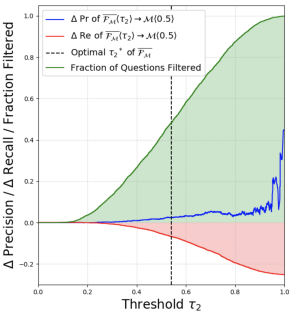
We observe that the filtering gains at the optimal ${\tau_2}^*$ are inversely correlated with the precision of $\mathcal{M}$. For example, for (12-layer $\mathcal{M}$, $\mathcal{F}$) at τ_1= 0.5, the Pr of SQuAD 1.1 and ASNQ is 88.7 and 74.6 respectively, and that of AQAD is significantly lower than ASNQ (due to real world noise). The % of questions filtered by $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}\langle\tau_2^*\rangle$ or $\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau_2^*\rangle$ increases in the order from 9$-$9.4% to 14.9$-$17.8%

			$\mathcal{M}$: 12-Layer Transformer $\mathcal{F}$: RoBERTa-Base					$\mathcal{M}$: 24-Layer Transformer $\mathcal{F}$: RoBERTa-Large				
Threshold τ_1 for $\mathcal{M}\rightarrow$			0.3	0.5	0.6	0.7	0.9	0.3	0.5	0.6	0.7	0.9
WikiQA	$\mathcal{M}\langle\tau_1\rangle$	Pr	84.4	87.1	88.4	88.4	97.1	92.1	92.2	92.0	92.1	95.9
		Re	80.2	75.3	68.7	63.0	28.0	86.0	83.1	80.2	77.0	57.2
	$\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	% Filter	4.1	2.9	3.7	7.0	42.8	0.8	3.3	6.6	7.8	18.9
		Δ Re	-2.4	- 0.8	-1.6	-2.5	- 3.7	-0.8	-2.0	- 4.1	- 5.0	-6.1
	$\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	% Filter	4.1	17.3	20.9	21.0	44.0	1.2	1.8	2.6	3.9	8.8
		Δ Re	-2.8	-10.3	- 9.0	-8.3	-7.4	-0.8	-0.8	-0.4	-2.1	-2.9
ASNQ	$\mathcal{M}\langle\tau_1\rangle$	Pr	68.2	74.6	77.2	79.6	90.5	75.7	79.6	81.9	84.5	92.9
		Re	48.7	41.1	36.1	28.9	10.0	61.1	54.4	49.3	42.7	20.7
	$\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	% Filter	7.8	17.8	29.8	54.2	83.8	0.2	10.5	15.6	29.8	66.2
		Δ Re	-1.8	-2.9	-4.7	-8.2	-3.9	0	- 3.2	-3.0	-7.4	-7.0
	$\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	% Filter	6.0	14.9	33.5	47.1	86.0	1.0	12.1	16.1	21.6	61.6
		Δ Re	-1.2	-2.7	-5.3	-6.4	-5.1	-0.1	-3.3	-3.5	-3.9	-6.3
SQuAD 1.1	$\mathcal{M}\langle\tau_1\rangle$	Pr	82.0	88.7	91.0	93.4	96.7	86.0	90.1	92.3	94.4	97.6
		Re	75.0	63.3	54.6	45.9	28.7	82.9	75.3	67.5	59.7	42.4
	$\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	% Filter	0	9.4	12.9	27.4	61.0	0	2.0	6.4	14.0	47.5
		Δ Re	0	- 2.3	-2.4	-4.8	-7.7	0	-0.5	-1.8	-3.8	-3.7
	$\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	% Filter	1.1	9.0	14.2	36.5	63.0	0	2.4	5.1	18.2	41.8
		Δ Re	-0.4	-2.3	-3.2	-8.2	-8.8	0	-0.5	-1.2	-5.5	-8.1
AQAD	$\mathcal{M}\langle\tau_1\rangle$	Pr	↑9.2	↑17.9	↑23.4	↑28.1	↑43.0	↑9.2	↑17.9	↑23.4	↑28.1	↑43.0
		Re	↓4.5	↓12.1	↓15.7	↓20.1	↓31.9	↓4.5	↓12.1	↓15.7	↓20.1	↓31.9
	$\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	% Filter	20.8	43.2	53.9	65.2	89.4	21.9	45.8	56.9	66.2	89.9
		Δ Re	-3.4	-5.0	-5.6	-6.0	-3.4	-3.2	-4.9	-5.5	-4.6	-3.0
	$\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	% Filter	17.8	48.2	59.8	75.7	91.7	15.2	48.7	57.3	72.0	93.8
		Δ Re	-2.8	-6.6	-7.5	-8.7	-3.8	-1.7	-5.7	-6.2	-7.0	-3.9

Table 1: Filtering gains and drop in recall for question filters operating at optimal filtering threshold τ_2^*. For a particular filter $\mathcal{F}$ operating with answer model $\mathcal{M}\langle\tau_1\rangle$, Δ Re refers to the difference in Recall of $\mathcal{F}\langle\tau_2{}^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$ and $\mathcal{M}\langle\tau_1\rangle$. % Filter refers to the % of questions preemptively discarded by $\mathcal{F}$. $\mathcal{M}\langle\tau_1\rangle$ results for AQAD are relative to $\mathcal{M}\langle 0\rangle$.



(a) ASNQ



(b) AQAD

Figure 5: Δ Pr/Δ Re plots and fraction of questions filtered on varying τ_2 for RoBERTa-Base $\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau_2\rangle\rightarrow\mathcal{M}\langle 0.5\rangle$ on ASNQ and AQAD datasets.

to 43.2−48.2%. The efficiency gain of our filters thus increases as the QA task becomes increasingly difficult (SQuAD 1.1 → ASNQ → AQAD). Furthermore, except for some cases on WikiQA, we observe that our question filters increase the precision of the system (for full table with ΔPr and ΔF1 refer Table 5 in Appendix F). This is in line with our observations in Fig. 2 and Kamath et al..

WikiQA (873 questions) is a very small dataset for efficiently distilling information from a transformer $\mathcal{M}$. Standard distillation (Hinton et al., 2015) often requires millions of training samples for efficient learning. To mitigate this, we extrapolate sequential fine-tuning as presented in (Garg et al., 2020) for learning question filters for WikiQA. We perform a two step learning of $\overline{\mathcal{F}_{\mathcal{M}}}$, $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$: first on ASNQ and then on WikiQA. The results for WikiQA in Table 1 correspond to this paradigm of training $\overline{\mathcal{F}_{\mathcal{M}}}$ and $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$, and demonstrate that our approach works to a reasonable level even on very small datasets. This also has implication towards shared semantics of question filters for models $\mathcal{M}$ trained on different datasets.

The drop in Re and filtering gains are contingent on the Pr/Re curve of $\mathcal{M}$ for the dataset. At higher thresholds (say τ_1=0.9), if the drop in recall due to $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$ or $\overline{\mathcal{F}_{\mathcal{M}}}$ at $\tau_2{}^*$ is more than desirable, then one can reduce the value of τ_2 down from $\tau_2{}^*$ by reducing the efficiency gains using plots like Fig. 5.

Comparison with Baselines: We also present results on optimal $\tau_2{}^*$ for $\mathcal{F}_{\text{W}}$ and $\mathcal{F}_{\text{C}}$ in Table 2 for ASNQ (complete results for all datasets are in Appendix F). When compared with the performance of $\overline{\mathcal{F}_{\mathcal{M}}}$ and $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$ for ASNQ in Table 1, both $\mathcal{F}_{\text{W}}$ and $\mathcal{F}_{\text{C}}$ perform inferior in terms of filtering performance. $\mathcal{F}_{\text{W}}$, which evaluates well-formedness of the questions from a human perspective, is unable to filter out any questions even when operating at its optimal threshold. This indicates that human-supervised filtering of ill-formed questions is sub-optimal from an efficiency perspective. $\mathcal{F}_{\text{C}}$ gets better performance than $\mathcal{F}_{\text{W}}$, but always trails $\overline{\mathcal{F}_{\mathcal{M}}}$ and $\mathcal{F}_{\mathcal{M}\langle\tau\rangle}$ either in terms of a smaller % of questions filtered or a larger drop in recall incurred.

Efficiency Gains from Filtering: Under simplifying assumptions, the computational resources required to answer questions within a fixed time budget over a fixed set of documents scales roughly linearly with the number of concurrent requests that need to be processed by Ω. We present a simple analysis on ASNQ (ignoring cost of retrieval $\mathcal{R}, \mathcal{S}$) considering 1000 questions (ASNQ has 400

τ_1 of $\mathcal{M}\downarrow$	$\mathcal{F}_{\text{W}}$		$\mathcal{F}_{\text{C}}$	
	Base	Large	Base	Large
0.3	0.3 / -0.1	0.1 / -0.2	0.2 / -0.1	0.3 / -0.2
0.5	0.9 / -0.4	0.1 / -0.1	1.0 / -0.2	0.5 / -0.3
0.6	0.9 / -0.4	0.1 / -0.2	22.3 / -5.0	2.7 / -0.8
0.7	1.0 / -0.2	0.2 / -0.1	38.5 / -6.2	7.2 / -1.3
0.9	0 / 0	0 / 0	84.4 / -4.8	62.5 / -8.1

Table 2: % Filter / Δ Recall results of baseline question filters $\mathcal{F}_{\text{W}}$ and $\mathcal{F}_{\text{C}}$ at optimal $\tau_2{}^*$ on ASNQ. Metrics are similar to those in Table 1. Base and Large refers to RoBERTa-Base and RoBERTa-Large question filters.

candidate answers/question) and a batch-size=100. $\mathcal{M}$ requires $1000*400/100$=4000 transformer forward passes (max_seq_length=128, standard for QA tasks due to long answers). On the other hand, max_seq_length=32 suffices for $\mathcal{F}$. Since inference latency of transformers scales roughly quadratic over input sequence length, 1 batch through $\mathcal{M}$ is 4^2=16 times slower than through $\mathcal{F}$. Assuming 20% question filtering by $\mathcal{F}$, $\mathcal{M}$ now only answers 800 questions (3200 forward passes of M), while adding $1000/100$=10 forward passes of $\mathcal{F}$. The %-cost reduction in time is $19.968\%\sim20\%$. We perform inference on ASNQ test-set on one V100-GPU (with $\overline{\mathcal{F}_{\mathcal{M}}}$ set to filter out 20% as per above) and observe latency dropping from 531.29s $\rightarrow$ 433.16s (18.47%, slightly lower than calculated 19.968% due to input/output overheads). The latency reduction can also translate to a reduction in the number of GPU compute resources required when performing inference in parallel. Furthermore, in practice, our filter will also provide cost/latency savings by not performing document retrieval for the filtered out questions.

5.6 Qualitative Analysis

In Table 3, we discuss some examples to highlight a few shortcomings of $\mathcal{F}$. Both $\mathcal{F}$,$\mathcal{M}$ can successfully filter out non-factual queries asking for opinions (examples 1, 2). Identifying popular entities like ("Jennifer Lopez", "Lakers", "horses") while training, $\mathcal{F}$ incorrectly assumes that a question composed of these entities will be answered by the system. While it may happen that due to unavailability of a web document having the exact answer, $\mathcal{M}$ might not answer the question (examples 3, 4). On the other hand, being unfamiliar with entities not encountered during training ("Ahsoka Tano","Mandalorian") or syntactically-complex questions, $\mathcal{F}$ preemptively might filter out questions which actually will be answered by $\mathcal{M}$ (examples 5, 6).

Question	$\mathcal{F}$	$\mathcal{M}$
1. What's your favorite movie series?	✗	✗
2. Where is the key to the building?	✗	✗
3. Was Jennifer Lopez a cheerleader for the Lakers?	✓	✗
4. What are two things that all horses have?	✓	✗
5. Which mayors of New York City had the name David?	✗	✓
6. Does Ahsoka Tano appear in the Mandalorian?	✗	✓

Table 3: Qualitative examples of questions along with the filtering decision of $\mathcal{M}\langle 0.5\rangle$ and $\overline{\mathcal{F}_{\mathcal{M}}}\langle 0.5\rangle$ (✓ / ✗ indicates clearing / failing the threshold).

6 Conclusion and Future Work

In this paper, we have presented a novel paradigm of training a question filter to capture the semantics of a QA system's answering capability by distilling the knowledge of the answer scores from it. Our experiments on three academic and one industrial QA benchmark show that the trained question models can estimate the Pr/Re curves of the QA system well, and can be used to effectively filter questions while only incurring a small drop in recall.

An interesting future work direction is to analyze the impact/behavior of the question filters in a cross-domain setting, where the training and testing corpora are from different domains. This would allow examining the transferability of the semantics learned by the question filters. A complementary future work direction could be knowledge distillation from a sophisticated answer verification module like (Rodriguez et al., 2019; Kamath et al., 2020; Zhang et al., 2021).

In addition to providing efficiency gains, the question filters could be used to qualitatively study the characteristics of questions that are likely to lead to low answer confidence scores. This can (i) help error analysis for improving the accuracy of QA systems, and (ii) be used for efficient sampling of training questions that are harder to be answered by the target QA system.

Our approach for training the question filters proposes the idea of partial-input knowledge distillation (e.g., using only questions instead of QA pairs). This concept can possibly be extended to other NLP problems for achieving compute efficiency gains, improved explainability (e.g., to what extent a partial-input influences model prediction) and qualitative analysis.

Acknowledgements

We thank the anonymous reviewers and meta-reviewer for their valuable suggestions. We thank Thuy Vu for developing and sharing the human annotated data used in the AQAD dataset.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2019. Learning to retrieve reasoning paths over wikipedia graph for question answering. *CoRR*, abs/1911.10470.

Weijie Bian, Si Li, Zhao Yang, Guang Chen, and Zhiqing Lin. 2017. A compare-aggregate model with dynamic-clip attention for answer selection. In *CIKM 2017*, CIKM '17, pages 1987–1990, New York, NY, USA. ACM.

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Andrea Gesmundo, Neil Houlsby, Wojciech Gajewski, and Wei Wang. 2017. Ask the right questions: Active question reformulation with reinforcement learning. *CoRR*, abs/1705.07830.

David Carmel and Elad Yom-Tov. 2010. Estimating the query difficulty for information retrieval. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, page 911, New York, NY, USA. Association for Computing Machinery.

Rishav Chakravarti and Avirup Sil. 2021. Towards confident machine reading comprehension.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Association for Computational Linguistics (ACL)*.

Danqi Chen and Wen-tau Yih. 2020. Open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 34–37, Online. Association for Computational Linguistics.

Jiecao Chen, Liu Yang, Karthik Raman, Michael Bendersky, Jung-Jung Yeh, Yun Zhou, Marc Najork, Danyang Cai, and Ehsan Emadzadeh. 2020. Dipair: Fast and accurate distillation for trillion-scale text matching and pair modeling. *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Zewei Chu, Mingda Chen, Jing Chen, Miaosen Wang, Kevin Gimpel, Manaal Faruqui, and Xiance Si. 2019. How to ask better questions? A large-scale multi-domain dataset for rewriting ill-formed questions. *CoRR*, abs/1911.09247.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618, Sofia, Bulgaria. Association for Computational Linguistics.

Manaal Faruqui and Dipanjan Das. 2018. Identifying well-formed natural language questions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 798–803, Brussels, Belgium. Association for Computational Linguistics.

Luke Gallagher, Ruey-Cheng Chen, Roi Blanco, and J. Shane Culpepper. 2019. Joint optimization of cascade ranking models. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 15–23, New York, NY, USA. Association for Computing Machinery.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7780–7788.

David Gondek, Adam Lally, Aditya Kalyanpur, J. William Murdock, Pablo Ariel Duboué, Lei Zhang, Yue Pan, Zhaoming Qiu, and Chris Welty. 2012. A framework for merging and ranking of answers in DeepQA. *IBM Journal of Research and Development*, 56(3/4):14:1–14:12.

S. Gupta, S. Lakra, and M. Kaur. 2020. Study on bert model for hate speech detection. In *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1–8.

Rujun Han, Luca Soldaini, and Alessandro Moschitti. 2021. Modeling context in answer sentence selection systems on a latency budget. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3005–3010, Online. Association for Computational Linguistics.

Claudia Hauff. 2010. Predicting the effectiveness of queries and retrieval systems. *SIGIR Forum*, 44(1):88.

Ben He and Iadh Ounis. 2004. Inferring query performance using pre-retrieval predictors. In *String Processing and Information Retrieval*, pages 43–54, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jiyin He, Martha Larson, and Maarten de Rijke. 2008. Using coherence-based measures to predict query difficulty. In *Advances in Information Retrieval*, pages 689–694, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Robin Jia and Wanze Xie. 2020. Know when to abstain: Calibrating question answering system under domain shift.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5684–5696, Online. Association for Computational Linguistics.

Jeongwoo Ko, Luo Si, and Eric Nyberg. 2007. A probabilistic framework for answer selection in question answering. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 524–531, Rochester, New York. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Ivano Lauriola and Alessandro Moschitti. 2021. Answer sentence selection using local and global context in transformer models. In *Advances in Information Retrieval*, pages 298–312, Cham. Springer International Publishing.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. FastBERT: a self-distilling BERT with adaptive inference time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6035–6044, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Xiaolu Lu, Soumajit Pramanik, Rishiraj Saha Roy, Abdalghani Abujabal, Yafang Wang, and Gerhard Weikum. 2019. Answering complex questions by joining multi-document evidence with quasi knowledge graphs. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 105–114, New York, NY, USA. Association for Computing Machinery.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. 2002. Is it the right answer? exploiting web redundancy for answer validation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 425–432, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Yoshitomo Matsubara, Thuy Vu, and Alessandro Moschitti. 2020. *Reranking for Efficient Transformer-Based Answer Selection*, page 1577–1580. Association for Computing Machinery, New York, NY, USA.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Knowledge guided text retrieval and reading for open domain question answering. *CoRR*, abs/1911.03868.

Josiane Mothe and Ludovic Tanguy. 2005. Linguistic features to predict query difficulty. In *In ACM SIGIR 2005 Workshop on Predicting Query Difficulty - Methods and Applications*.

Peng Qi, Xiaowen Lin, Leo Mehr, Zijian Wang, and Christopher D. Manning. 2019. Answering complex open-domain questions through iterative query generation. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Pedro Rodriguez, Shi Feng, Mohit Iyyer, He He, and Jordan L. Boyd-Graber. 2019. Quizbowl: The case for incremental question answering. *CoRR*, abs/1904.04792.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Denis Savenkov and Eugene Agichtein. 2016. When a knowledge base is not enough: Question answering over knowledge bases with external text data. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 235–244, New York, NY, USA. Association for Computing Machinery.

Sina J. Semnani and Manish Pandey. 2020. Revisiting the open-domain question answering pipeline.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 373–382, New York, NY, USA. ACM.

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017. Inter-weighted alignment network for sentence pair modeling. In *EMNLP 2017*, pages 1179–1189, Copenhagen, Denmark. Association for Computational Linguistics.

Luca Soldaini and Alessandro Moschitti. 2020. The cascade transformer: an application for efficient answer sentence selection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5697–5708, Online. Association for Computational Linguistics.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium. Association for Computational Linguistics.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242, Brussels, Belgium. Association for Computational Linguistics.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Shulong Tan, Zhixin Zhou, Zhaozhuo Xu, and Ping Li. 2019. On efficient retrieval of top similarity vectors. 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 5236–5246, Hong Kong, China. Association for Computational Linguistics.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Multi-cast attention networks for retrieval-based question answering and response prediction. *CoRR*, abs/1806.00778.

Quan Hung Tran, Tuan Lai, Gholamreza Haffari, Ingrid Zukerman, Trung Bui, and Hung Bui. 2018. The context-dependent additive recurrent neural net. In *NAACL 2018: Human Language Technologies, Volume 1 (Long Papers)*, pages 1274–1283, New Orleans, Louisiana. Association for Computational Linguistics.

Neeraj Varshney, Swaroop Mishra, and Chitta Baral. 2020. It's better to say "i can't answer" than answering incorrectly: Towards safety critical nlp systems.

Bingning Wang, Ting Yao, Qi Zhang, Jingfang Xu, Zhixing Tian, Kang Liu, and Jun Zhao. 2019. Document gated reader for open-domain question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 85–94, New York, NY, USA. Association for Computing Machinery.

Lidan Wang, Jimmy Lin, and Donald Metzler. 2011. A cascade ranking model for efficient ranked retrieval. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, page 105–114, New York, NY, USA. Association for Computing Machinery.

Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *(EMNLP-CoNLL) 2007*, pages 22–32, Prague, Czech Republic. Association for Computational Linguistics.

Qi Wang, Constantinos Dimopoulos, and Torsten Suel. 2016. Fast first-phase candidate generation for cascading rankers. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 295–304, New York, NY, USA. Association for Computing Machinery.

Shuohang Wang and Jing Jiang. 2016. A compare-aggregate model for matching text sequences. *CoRR*, abs/1611.01747.

Xuguang Wang, Linjun Shou, Ming Gong, Nan Duan, and Daxin Jiang. 2020. No answer is better than wrong answer: A reflection model for document level machine reading comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4141–4150, Online. Association for Computational Linguistics.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251, Online. Association for Computational Linguistics.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving question answering over incomplete KBs with knowledge-aware reader. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4258–4264, Florence, Italy. Association for Computational Linguistics.

Jie Yang, Claudia Hauff, Alessandro Bozzon, and Geert-Jan Houben. 2014. Asking the right question in collaborative qa systems. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, HT '14, page 179–189, New York, NY, USA. Association for Computing Machinery.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1253–1256, New York, NY, USA. Association for Computing Machinery.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *NAACL 2019*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering.

Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2019. A compare-aggregate model with latent clustering for answer selection. *CoRR*, abs/1905.12897.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. 2021. Knowing more about questions can help: Improving calibration in question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1958–1970, Online. Association for Computational Linguistics.

Ying Zhao, Falk Scholer, and Yohannes Tsegay. 2008. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Advances in Information Retrieval*, pages 52–64, Berlin, Heidelberg. Springer Berlin Heidelberg.

Appendix

A Dataset Details

All the datasets considered in this paper are in the English language. The dataset statistics for all the datasets are presented in Table 4.

WikiQA: A small-scale answer sentence selection dataset released by Yang et al. where the candidate answers are extracted from Wikipedia and the questions are derived from query logs of the Bing search engine. The search engine used for retrieving candidate answers is the Microsoft Bing web search engine. This dataset can be downloaded from the provided link [9]. This dataset has a subset of questions having no correct answer sentence ($all-$) or have only correct answer sentences ($all+$). The training is done by removing $all-$ questions, and the testing is done by removing both the $all-$ and $all+$ questions.

ASNQ: A large-scale answer sentence selection dataset released by Garg et al. where the candidate answers are from Wikipedia pages and the questions are from search queries of the Google search engine. ASNQ is a modified version of the Natural Questions (NQ) (Kwiatkowski et al., 2019) dataset by converting it from a MR dataset to an AS2 dataset. This is done by labelling sentences from the long answers which contain the short answer string as positive *correct* answer candidates and all others as negatives. This dataset can be downloaded from the provided link [10]. The dev split provided in the link is randomly divided into two equal components having 1336 questions each: one for validation, and the other for testing.

SQuAD1.1: A large-scale machine reading dataset released by Rajpurkar et al. where the questions have been written by crowdworkers and the answers are derived from Wikipedia articles. This requires predicting the start and end position (exact span) of the answer for a question from within the associated passage. Due to the hidden test set of SQuAD1.1 which is used for the leaderboard, we randomly divided the dev split into two components: one having 5266 questions (to be used for validation), and the other having 5267 questions (to be used for testing). All results on SQuAD in the paper are reported considering exact answer match. This dataset can be found at this link [11].

[9] http://aka.ms/WikiQA

[10] https://github.com/alexa/wqa_tanda

[11] https://rajpurkar.github.io/SQuAD-explorer/

Dataset	Train	Validation	Test
WikiQA	873	121	237
ASNQ	57,242	1,336	1,336
SQuAD 1.1	87,342	5,266	5,267
AQAD	1,000,000	50,000	5,000

Table 4: Dataset statistics providing exact details of number of questions in the train/validation/test split.

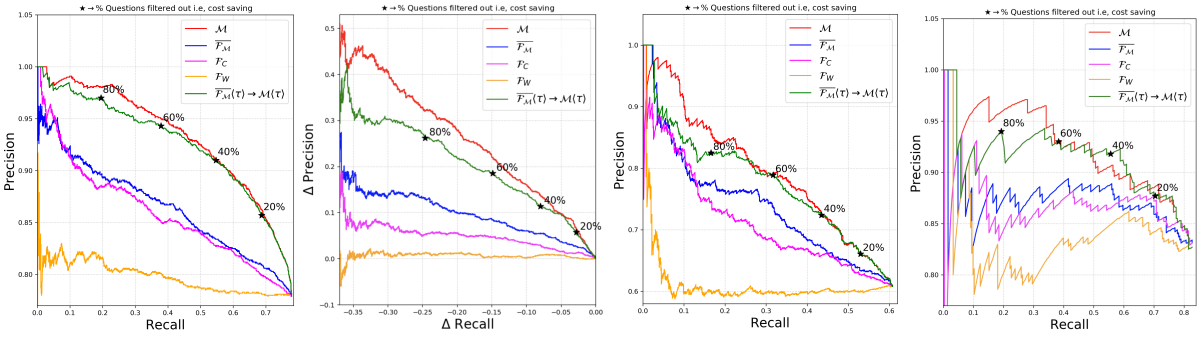
AQAD: A large scale internal industrial QA dataset derived from Alexa virtual assistant. Alexa QA Dataset (AQAD) contains 1 million and 50k questions in its train and dev. sets respectively, with their top answer and confidence scores as provided by the QA system. Note that the question answer pairs are without any human labels of correctness/incorrectness. The top answer is selected using an answer sentence selection model from hundreds of candidates that are retrieved from a large web-index ($\sim$ 1B web pages). For testing, we use 5000 questions (other than those in the train/dev. splits), each of which is human annotated with a label corresponding to the top answer from the QA system being correctly or incorrect. For learning the correctness filter $\mathcal{F}_C$ baseline on AQAD, we use an additional annotated split of 2,500 questions other than the train/dev./test splits.

B Model Details

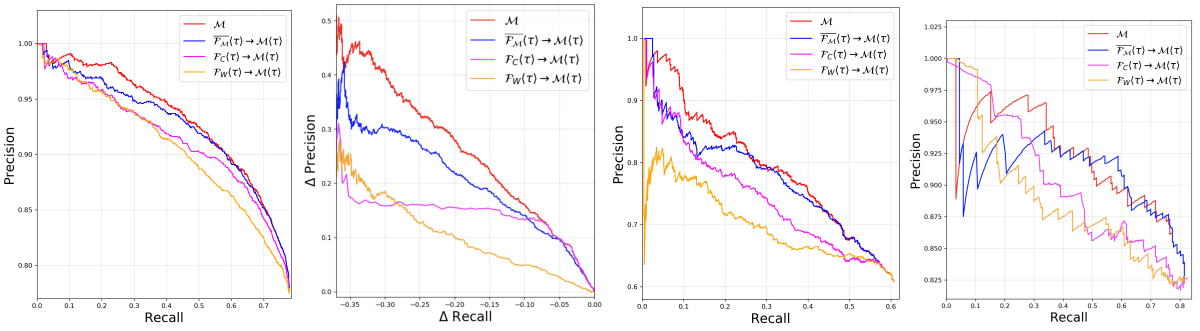
For each of the datasets we describe the details of the answer models $\mathcal{M}$ for reproducibility purposes:

- **WikiQA:** We consider the TANDA model checkpoints released by Garg et al. which are trained using sequential fine-tuning and are the state-of-the-art QA models for WikiQA. Specifically we consider the RoBERTa-Base 12-layer model first trained on ASNQ and then on WikiQA and the RoBERTa-Large-MNLI 24-layer model first trained on ASNQ and then on WikiQA. Baseline accuracy for the 12 and 24-layer model $\mathcal{M}$ on the test split is 82.7% and 91.8% respectively.

- **ASNQ:** We consider the RoBERTa-Base and RoBERTa-Large-MNLI model checkpoints which have been fine-tuned on the training set of ASNQ for 3 epochs using a learning rate of 2e-5 Adam and have been released by Garg et al.. Baseline accuracy for the 12 and 24-layer model $\mathcal{M}$ on the test split is 60.8% and 69.2% respectively.



(a) SQuAD 1.1 (b) AQAD (c) ASNQ (d) WikiQA



(e) SQuAD 1.1 (f) AQAD (g) ASNQ (h) WikiQA

Figure 6: Pr/Re curves for filters ($\overline{\mathcal{F}_{\mathcal{M}}}$, $\mathcal{F}_{\mathrm{W}}$, $\mathcal{F}_{\mathrm{C}}$) and answer model $\mathcal{M}$ are presented in (a)-(d). Pr/Re curves for filters jointly operating with $\mathcal{M}$, i.e, $\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau\rangle{\rightarrow}\mathcal{M}\langle\tau\rangle$, $\mathcal{F}_{\mathrm{W}}\langle\tau\rangle{\rightarrow}\mathcal{M}\langle\tau\rangle$, $\mathcal{F}_{\mathrm{C}}\langle\tau\rangle{\rightarrow}\mathcal{M}\langle\tau\rangle$ are presented in (e)-(h). For AQAD we show ΔPr/ΔRe w.r.t $\mathcal{M}\langle 0\rangle$.

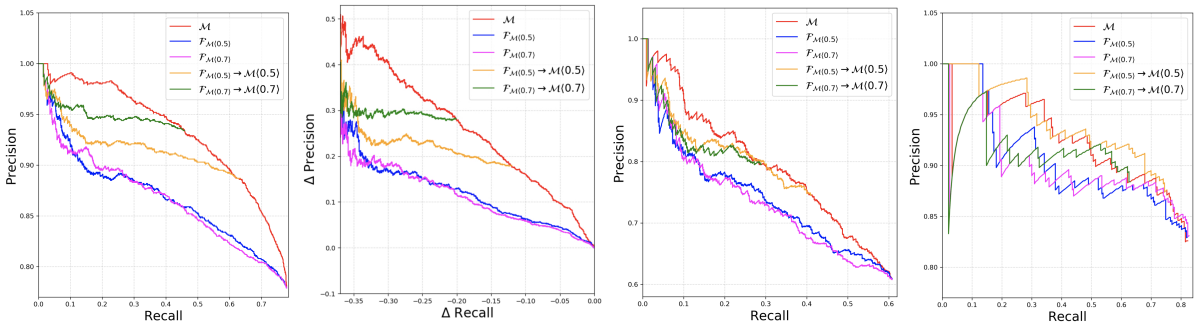
- **SQuAD1.1:** We consider the BERT-Base uncased and BERT-Large uncased with whole word masking model variants and fine-tune them on the training set of SQuAD1.1 for 3 epochs with a standard learning rate of 2e-5 Adam and learning rate warm-up set for the first 5% of training steps. Baseline accuracy for the 12 and 24-layer model $\mathcal{M}$ on the test split is 77.9% and 84.0% respectively.

- **AQAD:** We consider the ELECTRA-Base (Clark et al., 2020) model and perform sequential fine-tuning using TANDA (Garg et al., 2020) by a first round of fine-tuning on ASNQ for 3 epochs with a learning rate of 2e-5 Adam (learning rate warm-up of 5%), followed by a second round of fine-tuning for 3 epochs with a learning rate of 2e-6 Adam (learning rate warm-up of 5%). Baseline accuracy is not disclosed since the data is internal.

C Experimental Details

Training Details: All computations are performed on NVIDIA Telsa V100 GPUs with a batch-size of 128. For training a question filter, we train $\mathcal{F}$ using the proposed loss objectives for 3 epochs on the training split of the dataset using a standard learning rate of 2e-5 Adam (with learning rate warm-up set for the first 5% of the training steps). RoBERTa-Base and RoBERTa-Large question filters are trained corresponding to 12 and 24 layer answer models $\mathcal{M}$ respectively. For AQAD, both the RoBERTa-Base and RoBERTa-Large question filters are trained corresponding to the ELECTRA-Base answer model $\mathcal{M}$. For WikiQA, the question filters are trained by sequential training: first on ASNQ for 3 epochs using a standard learning rate of 2e-5 Adam (with learning rate warm-up set for the first 5% of the training steps), and then on WikiQA for 3 epochs with a learning rate of 3e-6 (with learning rate warm-up set for first 5% of the training steps). The baseline classifier for wellformedness $\mathcal{F}_{\mathrm{W}}$ and correctness of answering a question $\mathcal{F}_{\mathrm{C}}$ are also trained by fine-tuning the RoBERTa-Base and Large models for 3 epochs using a standard learning rate of 2e-5 (with learning rate warm-up set for the first 5% of the training steps). As mentioned in Appendix A, we use an additional annotated data split containing 250k QA pairs (2.5k questions) for training $\mathcal{F}_{\mathrm{C}}$ on AQAD.

Validation Strategy: For computing the optimal threshold ${\tau_2}^*$ of a question filter $\mathcal{F}$ as described in



(a) SQuAD 1.1 (b) AQAD (c) ASNQ (d) WikiQA

Figure 7: Pr/Re curves for filters ($\mathcal{F}_{\mathcal{M}\langle 0.5\rangle}$, $\mathcal{F}_{\mathcal{M}\langle 0.7\rangle}$), answer model $\mathcal{M}$ and operating configurations ($\mathcal{F}_{\mathcal{M}\langle 0.5\rangle}{\rightarrow}\mathcal{M}\langle 0.5\rangle$, $\mathcal{F}_{\mathcal{M}\langle 0.7\rangle}{\rightarrow}\mathcal{M}\langle 0.7\rangle$) are presented. For AQAD we show ΔPr/ΔRe w.r.t $\mathcal{M}\langle 0\rangle$.

Section 5.5, we use the dev. split of the datasets to find $\tau_2{}^* \in [0, 1]$ such that $\mathcal{F}\langle\tau_2^*\rangle$ obtains the highest F1-score corresponding to the binary decision of answering or abstaining by $\mathcal{M}\langle\tau_1\rangle$. We present concrete results corresponding to 5 different operating thresholds τ_1 of the answer model $\mathcal{M}$: $\{0.3, 0.5, 0.6, 0.7, 0.9\}$. At each τ_1 for $\mathcal{M}$, we consider all 4 different possible question filters: our answer-model distilled question filter with regression head $\overline{\mathcal{F}_{\mathcal{M}}}$, our answer-model distilled question filter with classification head $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$, the correctness question filter $\mathcal{F}_{\text{C}}$ and the well-formedness question filter $\mathcal{F}_{\text{W}}$. For each of these four filters, we independently optimise $\tau_2{}^* \in [0, 1]$ corresponding to best F1 filtering of $\mathcal{M}\langle\tau_1\rangle$.

Code: The code for training our answer-model distilled question filters can be accessed at https://github.com/alexa/wqa-question-filtering.

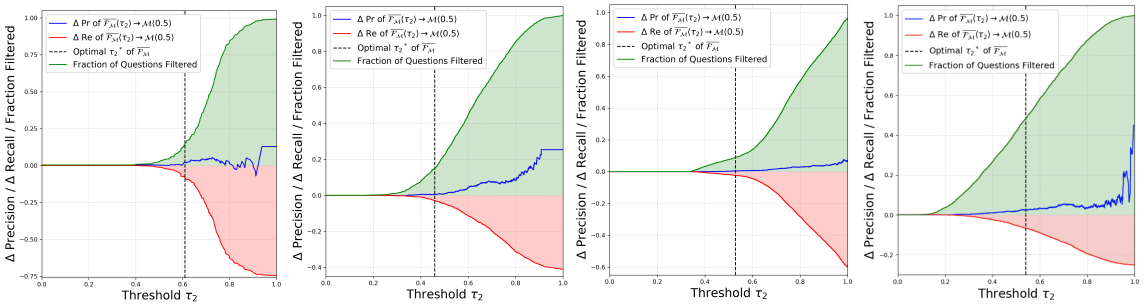
D Complete Graphs on Approximating Pr/Re of $\mathcal{M}$

We present Pr/Re curves of $\mathcal{M}$ by varying τ_1 (i.e, $\mathcal{M}\langle\tau_1\rangle$) and that of filters $\mathcal{F}$: $\{\overline{\mathcal{F}_{\mathcal{M}}}, \mathcal{F}_{\text{C}}, \mathcal{F}_{\text{W}}\}$ by varying τ_2 (i.e, $\mathcal{F}\langle\tau_2\rangle$) on the dataset test splits in Fig. 6 (a)-(d). We also present Pr/Re curves comparing the filters $\mathcal{F}$: $\{\overline{\mathcal{F}_{\mathcal{M}}}, \mathcal{F}_{\text{C}}, \mathcal{F}_{\text{W}}\}$ when operating jointly with the answer model $\mathcal{M}$ at the same threshold τ, i.e, $\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau\rangle{\rightarrow}\mathcal{M}\langle\tau\rangle$, $\mathcal{F}_{\text{C}}\langle\tau\rangle{\rightarrow}\mathcal{M}\langle\tau\rangle$ and $\mathcal{F}_{\text{W}}\langle\tau\rangle{\rightarrow}\mathcal{M}\langle\tau\rangle$ on the dataset test splits in Fig. 6 (e)-(h). As visible from the graphs, our filter $\overline{\mathcal{F}_{\mathcal{M}}}$ is able to better approximate the Pr/Re of $\mathcal{M}$ when operating independently of $\mathcal{M}$ (Fig. 6 (a)-(d)) as well as when operating jointly with $\mathcal{M}$ at a non-zero threshold (Fig. 6 (e)-(h)). Note that the trained answer models on WikiQA (which has a very small test set having only 237 samples) are very poorly calibrated. This is visible from the shape of the Pr/Re curve of $\mathcal{M}$ in Fig. 6 (d),(h). Interestingly, even for such a poorly calibrated answer model, our filter $\overline{\mathcal{F}_{\mathcal{M}}}$ is still able to approximate the Pr/Re of $\mathcal{M}$ better than the baselines $\mathcal{F}_{\text{C}}$ and $\mathcal{F}_{\text{W}}$. This illustrates the validity of our technique even for very small datasets (few-shot setting).

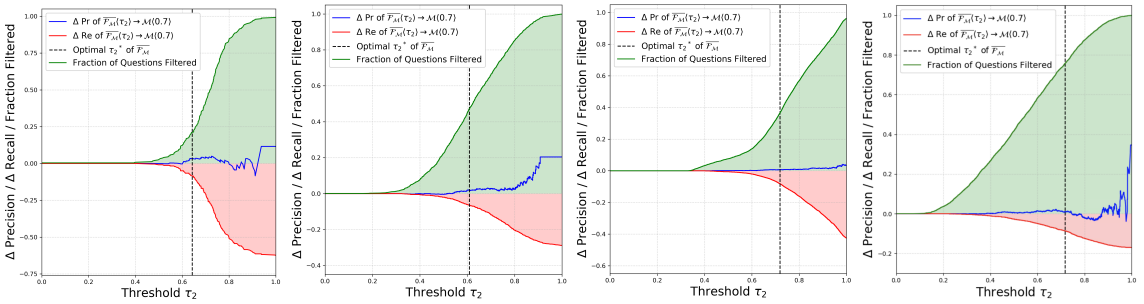
Our classification-head filter $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$ is trained specific to a threshold τ_1 of $\mathcal{M}$. Since each point in the Pr/Re graph of $\mathcal{M}$ corresponds to a different threshold $\tau_1 \in [0, 1]$, for fair comparison in Fig. 6, we will need to train several $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$ for every $\tau_1 \in [0, 1]$ which is unfeasible. To show how the classification head filters can approximate the Pr/Re of $\mathcal{M}$, we arbitrarily select two thresholds $\tau_1{=}\{0.5, 0.7\}$ and plot the Pr/Re curves of the two classification head filters $\mathcal{F}_{\mathcal{M}\langle 0.5\rangle}$ and $\mathcal{F}_{\mathcal{M}\langle 0.7\rangle}$ in Fig. 7. We also plot the operating configurations for these filters at the corresponding τ_1 for $\mathcal{M}$, i.e, $\mathcal{F}_{\mathcal{M}\langle 0.5\rangle}{\rightarrow}\mathcal{M}\langle 0.5\rangle$ and $\mathcal{F}_{\mathcal{M}\langle 0.7\rangle}{\rightarrow}\mathcal{M}\langle 0.7\rangle$ in Fig. 7.

E Complete Graphs on Varying Threshold τ_2 of $\mathcal{F}$

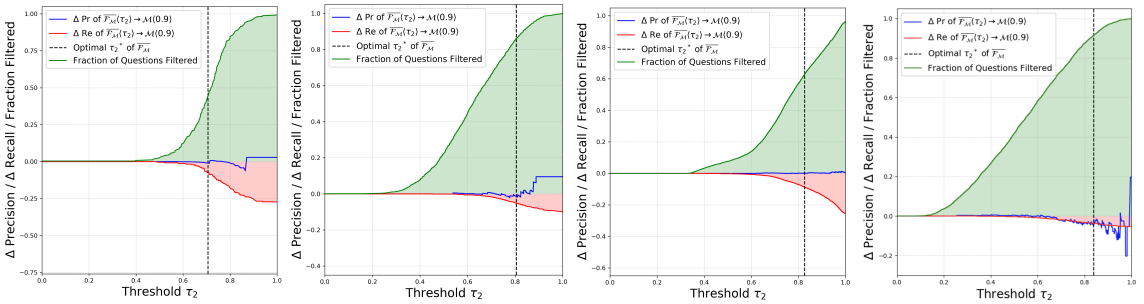
We present the variation of the fraction of questions filtered by $\mathcal{F}$ along with the change in Pr/Re of $\widehat{\Omega}$ on varying τ_2 for all the datasets in Fig. 8. We present plots for three different operating thresholds $\tau_1{=}\{0.5, 0.7, 0.9\}$ of the answer model $\mathcal{M}$. For a dataset, since $\overline{\mathcal{F}_{\mathcal{M}}}$ is trained independent of any threshold τ_1, the fraction of filtered questions would remain the same as we vary τ_2 even at different values of τ_1. Using these graphs, one can choose the desired operating point for the filter corresponding to how much efficiency gain is desired and how much drop in recall can be tolerated.



(a) WikiQA, τ_1=0.5 (b) ASNQ, τ_1=0.5 (c) SQuAD 1.1, τ_1=0.5 (d) AQAD, τ_1=0.5



(e) WikiQA, τ_1=0.7 (f) ASNQ, τ_1=0.7 (g) SQuAD 1.1, τ_1=0.7 (h) AQAD, τ_1=0.7



(i) WikiQA, τ_1=0.9 (j) ASNQ, τ_1=0.9 (k) SQuAD 1.1, τ_1=0.9 (l) AQAD, τ_1=0.9

Figure 8: Δ Pr/ Δ Re and fraction of questions filtered on varying τ_2 for RoBERTa-Base $\overline{\mathcal{F}_{\mathcal{M}}}$ on all datasets for three different τ_1 for $\mathcal{M}$: $\{0.5, 0.7, 0.9\}$.

F Complete Results for Optimal τ_2 of $\mathcal{F}$

We present the complete empirical results of our filters: $\overline{\mathcal{F}_{\mathcal{M}}}$ and $\mathcal{F}_{\mathcal{M}\langle\tau\rangle}$ at different thresholds τ_1 of $\mathcal{M}$ in Table 5. We evaluate the % of questions filtered out by $\mathcal{F}\langle\tau_2^*\rangle$ (efficiency gains) and the resulting drop in Precision, Recall and question-answering F1 score of $\mathcal{F}\langle\tau_2^*\rangle \rightarrow \mathcal{M}\langle\tau_1\rangle$ from $\mathcal{M}\langle\tau_1\rangle$ on the test split of the dataset. For each dataset and model $\mathcal{M}$, we train one regression head question filter $\overline{\mathcal{F}_{\mathcal{M}}}$ and five classification head question filters $\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}$: one at every threshold τ_1 for $\mathcal{M} \in \{0.3, 0.5, 0.6, 0.7, 0.9\}$. The optimal filtering threshold $\tau_2{}^*$ is computed using the validation strategy described in Appendix C. For the regression head $\overline{\mathcal{F}_{\mathcal{M}}}$, the optimal $\tau_2{}^*$ is calculated independently for every τ_1 of $\mathcal{M}$.

Additionally we present the complete empirical results on all datasets corresponding to the optimal filtering threshold $\tau_2{}^*$ for the baseline question filters: $\mathcal{F}_{\text{C}}$ and $\mathcal{F}_{\text{W}}$ in Table 6. We observe that both $\mathcal{F}_{\text{W}}$ and $\mathcal{F}_{\text{C}}$ perform inferior in terms of filtering performance to our filters $\overline{\mathcal{F}_{\mathcal{M}}}$ and $\mathcal{F}_{\mathcal{M}\langle\tau\rangle}$. Except for higher thresholds on AQAD, the well-formedness filter $\mathcal{F}_{\text{W}}$ is unable to filter out a sizable fraction of questions even when operating at $\tau_2{}^*$ which indicates that human-supervised filtering of ill-formed questions is sub-optimal from an efficiency perspective. $\mathcal{F}_{\text{C}}$ gets better performance than $\mathcal{F}_{\text{W}}$, but always trails $\overline{\mathcal{F}_{\mathcal{M}}}$ and $\mathcal{F}_{\mathcal{M}\langle\tau\rangle}$ either in terms of a smaller % of questions filtered or a larger drop in recall incurred.

			$\mathcal{M}$: 12-Layer Transformer Architecture, $\mathcal{F}$: RoBERTa-Base					$\mathcal{M}$: 24-Layer Transformer Architecture, $\mathcal{F}$: RoBERTa-Large				
	Threshold τ_1 for $\mathcal{M}\rightarrow$		0.3	0.5	0.6	0.7	0.9	0.3	0.5	0.6	0.7	0.9
WikiQA	$\mathcal{M}\langle\tau_1\rangle$	Pr / Re	84.4 / 80.2	87.1 / 75.3	88.4 / 68.7	88.4 / 63.0	97.1 / 28.0	92.1 / 86.0	92.2 / 83.1	92.0 / 80.2	92.1 / 77.0	95.9 / 57.2
		F1	82.2	80.8	77.3	73.6	43.5	89.9	87.4	85.7	83.9	71.7
	$\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	Δ Pr / Δ Re	-0.3 / -2.4	+0.3 / - 0.8	-0.3 / -1.6	+0.7 / -2.5	-0.4 / - 3.7	-0.1 / -0.8	+0.3 / -2.0	+0.5 / - 4.1	-0.5 / - 5.0	+0.2 / -6.1
		% Filter / Δ F1	4.1 / -0.9	2.9 / -0.4	3.7 / -1.1	7.0 / -1.5	42.8 / - 4.7	0.8 / -0.4	3.3 / - 1.0	6.6 / - 2.2	7.8 / - 3.3	18.9 / - 4.1
	$\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	Δ Pr / Δ Re	+0.7 / -2.8	+2.3 / -10.3	+2.8 / - 9.0	+3.3 / -8.3	-0.9 / -7.4	-0.1 / -0.8	0 / -0.8	-0.1 / -0.4	-0.2 / -2.1	-0.2 / -2.9
		% Filter / Δ F1	4.1 / - 1.1	17.3 / - 5.0	20.9 / - 5.1	21.0 / -5.1	44.0 / - 9.6	1.2 / -1.4	1.8 / -0.4	2.6 / -0.3	3.9 / -1.4	8.8 / -2.4
ASNQ	$\mathcal{M}\langle\tau_1\rangle$	Pr / Re	68.2 / 48.7	74.6 / 41.1	77.2 / 36.1	79.6 / 28.9	90.5 / 10.0	75.7 / 61.1	79.6 / 54.4	81.9 / 49.3	84.5 / 42.7	92.9 / 20.7
		F1	56.8	53.0	49.2	42.4	18.0	67.6	64.6	61.5	56.7	33.9
	$\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	Δ Pr / Δ Re	+0.9 / -1.8	+1.6 / -2.9	+1.6 / -4.7	+2.5 / -8.2	+0.6 / -3.9	0 / 0	+1.3 / - 3.2	+1.5 / -3.0	+1.8 / -7.4	+2.4 / -7.0
		% Filter / Δ F1	7.8 / -0.9	17.8 / -2.1	29.8 / -4.3	54.2 / -9.3	83.8 / -6.6	0.2 / 0	10.5 / -1.9	15.6 / -2.0	29.8 / -6.6	66.2 / -9.9
	$\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	Δ Pr / Δ Re	+0.7 / -1.2	+0.6 / -2.7	+1.4 / -5.3	+1.9 / -6.4	-1.3 / -5.1	+0.2 / -0.1	+0.8 / -3.3	+1.0 / -3.5	+1.4 / -3.9	+2.2 / -6.3
		% Filter / Δ F1	6.0 / -0.4	14.9 / -2.2	33.5 / -4.9	47.1 / -7.1	86.0 / -8.7	1.0 / 0	12.1 / -2.1	16.1 / -2.5	21.6 / -3.2	61.6 / -8.9
SQuAD 1.1	$\mathcal{M}\langle\tau_1\rangle$	Pr / Re	82.0 / 75.0	88.7 / 63.3	91.0 / 54.6	93.4 / 45.9	96.7 / 28.7	86.0 / 82.9	90.1 / 75.3	92.3 / 67.5	94.4 / 59.7	97.6 / 42.4
		F1	78.3	73.9	68.3	61.6	44.3	84.4	82.0	78.0	73.1	59.1
	$\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	Δ Pr / Δ Re	0 / 0	+0.3 / - 2.3	+0.4 / -2.4	+0.6 / -4.8	+0.1 / -7.7	0 / 0	+0.1 / -0.5	+0.3 / -1.8	+0.5 / -3.8	+0.3 / -3.7
		% Filter / Δ F1	0 / 0	9.4 / -1.5	12.9 / -1.9	27.4 / -4.4	61.0 / -9.8	0 / 0	2.0 / -0.2	6.4 / -1.1	14.0 / -2.7	47.5 / -3.7
	$\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	Δ Pr / Δ Re	+0.2 / -0.4	+0.5 / -2.3	+0.6 / -3.2	+0.7 / -8.2	+0.2 / -8.8	0 / 0	+0.1 / -0.5	+0.3 / -1.2	+0.7 / -5.5	+0.3 / -8.1
		% Filter / Δ F1	1.1 / -0.1	9.0 / -1.4	14.2 / -2.5	36.5 / -7.8	63.0 / -10.3	0 / 0	2.4 / -0.3	5.1 / -0.7	18.2 / -4.1	41.8 / -8.3
AQAD	$\mathcal{M}\langle\tau_1\rangle$	Pr / Re	↑9.2 / ↓4.5	↑17.9 / ↓12.1	↑23.4 / ↓15.7	↑28.1 / ↓20.1	↑43.0 / ↓31.9	↑9.2 / ↓4.5	↑17.9 / ↓12.1	↑23.4 / ↓15.7	↑28.1 / ↓20.1	↑43.0 / ↓31.9
	$\mathcal{F}_{\mathcal{M}\langle\tau_1\rangle}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	Δ Pr / Δ Re	+1.3 / -3.4	+1.9 / -5.0	+2.0 / -5.6	+1.2 / -6.0	-1.9 / -3.4	+1.8 / -3.2	+2.5 / -4.9	+2.3 / -5.5	+1.8 / -4.6	+1.9 / -3.0
		% Filter / Δ F1	20.8 / -2.1	43.2 / -4.8	53.9 / -6.4	65.2 / -8.0	89.4 / -6.2	21.9 / -1.8	45.8 / -4.6	56.9 / -6.3	66.2 / -7.4	89.9 / -5.4
	$\overline{\mathcal{F}_{\mathcal{M}}}\langle\tau_2^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$	Δ Pr / Δ Re	+1.5 / -2.8	+2.5 / -6.6	+2.1 / -7.5	+1.1 / -8.7	-4.1 / -3.8	+1.3 / -1.7	+2.7 / -5.7	+1.9 / -6.2	+2.3 / -7.0	+0.2 / -3.9
		% Filter / Δ F1	17.8 / -1.6	48.2 / - 6.5	59.8 / -8.9	75.7 / -12.2	91.7 / -7.0	15.2 / -0.8	48.7 / -5.4	57.3 / -7.2	72.0 / -9.5	93.8 / -7.1

Table 5: Results showing effectiveness of question filtering. For each dataset and model $\mathcal{M}$, we train 6 question filters: five $\mathcal{F}_{\mathcal{M}\langle i\rangle}$'s for $i \in \{0.3, 0.5, 0.6, 0.7, 0.9\}$ and one $\overline{\mathcal{F}_{\mathcal{M}}}$. For a particular filter $\mathcal{F}$ operating with $\mathcal{M}\langle\tau_1\rangle$, Δ (Pr/Re/F1) refers to the difference in (Pr/Re/F1) of $\mathcal{F}\langle\tau_2{}^*\rangle\rightarrow\mathcal{M}\langle\tau_1\rangle$ and $\mathcal{M}\langle\tau_1\rangle$. % Filter refers to the % of questions preemptively discarded by the question filter. $\mathcal{M}\langle\tau_1\rangle$ results for AQAD are relative to $\mathcal{M}\langle 0\rangle$.

	Threshold τ_1 for $\mathcal{M}\rightarrow$		0.3	0.5	0.6	0.7	0.9
WikiQA	$\mathcal{F}_{\text{W}}$	RoBERTa-Base	0 / 0	0.2 / -0.1	0 / 0	0.2 / -0.1	0.8 / -0.8
		RoBERTa-Large	0 / 0	0 / 0	0.2 / -0.1	0.3 / -0.2	0 / 0
	$\mathcal{F}_{\text{C}}$	RoBERTa-Base	4.1 / -1.2	1.5 / -0.8	2.1 / -0.8	4.1 / -0.9	15.6 / -1.7
		RoBERTa-Large	0.8 / -0.8	1.3 / -0.9	0.8 / -0.9	1.2 / -1.1	2.8 / -0.6
ASNQ	$\mathcal{F}_{\text{W}}$	RoBERTa-Base	0.3 / -0.1	0.9 / -0.4	0.9 / -0.4	1.0 / -0.2	0 / 0
		RoBERTa-Large	0.1 / -0.2	0.1 / -0.1	0.1 / -0.2	0.2 / -0.1	0 / 0
	$\mathcal{F}_{\text{C}}$	RoBERTa-Base	0.2 / -0.1	1.0 / -0.2	22.3 / -5.0	38.5 / -6.2	84.4 / -4.8
		RoBERTa-Large	0.3 / -0.2	0.5 / -0.3	2.7 / -0.8	7.2 / -1.3	62.5 / -8.1
SQuAD 1.1	$\mathcal{F}_{\text{W}}$	RoBERTa-Base	0 / 0	0.4 / -0.1	0.5 / -0.2	0.7 / -0.3	0.2 / -0.2
		RoBERTa-Large	0.1 / -0.1	0.3 / -0.2	0.3 / -0.4	0.5 / -0.2	0.3 / -0.4
	$\mathcal{F}_{\text{C}}$	RoBERTa-Base	0 / 0	0.3 / -0.1	28.0 / -9.1	34.3 / -8.7	59.7 / -9.4
		RoBERTa-Large	0 / 0	0.8 / -0.3	0.8 / -0.3	2.6 / -1.0	31.0 / -8.5
AQAD	$\mathcal{F}_{\text{W}}$	RoBERTa-Base	0 / 0	4.8 / -0.7	4.2 / -0.4	15.7 / -1.9	32.6 / -2.5
		RoBERTa-Large	0 / 0	4.3 / -1.1	4.4 / -0.8	15.6 / -2.3	29.6 / -3.2
	$\mathcal{F}_{\text{C}}$	RoBERTa-Base	4.5 / -0.8	18.5 / -3.1	38.5 / -5.8	38.9 / -4.5	82.3 / -3.6
		RoBERTa-Large	3.1 / -0.4	20.2 / -3.3	36.1 / -5.4	37.2 / -4.3	80.9 / -3.4

Table 6: Table presenting performance of baseline question filters $\mathcal{F}_{\text{W}}$ and $\mathcal{F}_{\text{C}}$ on all four datasets corresponding to their optimal operating threshold $\tau_2{}^*$.