Question Answering over Electronic Devices: A New Benchmark Dataset and a Multi-Task Learning based QA Framework

Abhilash Nandy[♠] **Soumya Sharma**[♠] **Shubham Maddhashiya**[♠] **Kapil Sachdeva**[♣]
Pawan Goyal[♠] **Niloy Ganguly**[♠♢]

[♠]Indian Institute of Technology, Kharagpur [♣]Samsung Research Institute, Delhi
[♢] L3S Research Center, Leibniz Universität Hannover

Abstract

Answering questions asked from instructional corpora such as E-manuals, recipe books, etc., has been far less studied than open-domain factoid context-based question answering. This can be primarily attributed to the absence of standard benchmark datasets. In this paper we meticulously create a large amount of data connected with E-manuals and develop suitable algorithm to exploit it. We collect **E-Manual Corpus**, a huge corpus of 307,957 E-manuals and pretrain RoBERTa on this large corpus. We create various benchmark QA datasets which include question answer pairs curated by experts based upon two E-manuals, real user questions from Community Question Answering Forum pertaining to E-manuals etc. We introduce **EMQAP (E-Manual Question Answering Pipeline)** that answers questions pertaining to electronics devices. Built upon the pretrained RoBERTa, it harbors a supervised multi-task learning framework which efficiently performs the dual tasks of identifying the section in the E-manual where the answer can be found and the exact answer span within that section. For E-Manual annotated question-answer pairs, we show an improvement of about 40% in ROUGE-L F1 scores over the most competitive baseline. We perform a detailed ablation study and establish the versatility of EMQAP across different circumstances. The code and datasets are shared at https://github.com/abhi1nandy2/EMNLP-2021-Findings, and the corresponding project website is https://sites.google.com/view/emanualqa/home.

1 Introduction

An E-Manual, or Electronic Manual, is a document that provides technical support to the consumers of a product by giving instructions and procedures to operate the device along with know-how of its specifications. It is often difficult to find the relevant instructions from an E-manual; hence, an automated question answering support to use the information present in the E-manual effectively would be of great help.

E-Manuals typically provide lengthy instructions structured in a sequential fashion explaining various uses of a device. This often poses a challenge in building a question answering system because the answer to a question may come from multiple disjointed portions within a section of the E-Manual. Due to the instructional nature of E-Manuals, we also find that often adjacent instructions are not related to each other but may be related to a parental instruction leading to long-range dependencies in context. This, therefore, deems a **domain-specific natural language understanding** which may, in turn, suffer from lack of domain-specific labeled data (Araci, 2019) and presence of formal syntax in the corpus (Beltagy et al., 2019; Chalkidis et al., 2020). These challenges have led recent works to pre-train the state-of-the-art transformer models on unlabelled domain-specific corpora (Lee et al., 2020; Araci, 2019; Beltagy et al., 2019; Chalkidis et al., 2020). Inspired by such works, we painstakingly collect **E-Manual Corpus: a huge corpus of 307,957 E-manuals**[1] and pre-train the transformer-based language model, RoBERTa_BASE[2] on the corpus (Section 3.1).

A **question answering system** needs to select the relevant section of the E-Manual, which contains the answer to the given question (**section retrieval (SR)**) and subsequently, extract the answer from that relevant section (**answer retrieval (AR)**). There are currently four main types of approaches in state-of-the-art literature that utilize the SR and AR systems (1). Chen et al. (2017) uses a two-stage training pipeline where the SR model consists of an unsupervised Information Retrieval (IR) method like TF-IDF or BM25, followed by an

[1] www.manualsonline.com

[2] Note that, in this paper, unless otherwise specified, 'RoBERTa' would just mean 'RoBERTa_BASE'

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4600–4609
November 7–11, 2021.

extractive AR model; (2) an end-to-end learning setup of SR cascaded by AR (Guu et al., 2020; Lee et al., 2019); (3) single-span (Rajpurkar et al., 2016) or multi-span (Zhu et al., 2020; Segal et al., 2020) answers given questions and corresponding candidate contexts as inputs and (4) a Multi-task Learning (MTL) Framework, where SR and AR are the two underlying tasks (Nishida et al., 2018); Nishida et al. (2018) performs MTL using separate SR and AR pipelines sharing feature extraction layers. The simultaneous training of SR and AR using MTL helps the model build a combined and hierarchical understanding of Question Answering at a global (section) and a local (sentence/token) level. However, these methods apply a span-based selection approach for extracting answers, whereas the answers to questions on E-Manuals are usually non-contiguous; hence while we principally use this **multi-task learning (MTL)** framework, we make some customization to accommodate the peculiarity of the data.

Summing up, the paper makes the following contributions: **(1)** Since no data is available for the E-Manual domain, we create a huge corpus for pre-training containing 307,957 E-Manuals known as the E-Manual Corpus. **(2)** Since no QA dataset is available for this domain, we apply multi-pronged strategy to create a large enough corpus of Question Answering (QA) datasets: two datasets **manually annotated by experts** containing **904 and 950 questions** respectively, and another collected from **Amazon Question Answering Forum containing 1,028 questions** and a set of **10 question-answer pairs for 40 different devices each** (Section 2). **(3)** EMQAP (E-Manual Question Answering Pipeline) develops on two basic pillars - a domain-specific **pre-trained RoBERTa architecture** and a **multi-task learning framework**.

In the next section we discuss in detail the different types of data rigorously created. The system design is discussed in detail in Section 3, followed by the experimental results in Section 4. The experimental results emphatically establish that the performance of EMQAP is way superior to its nearest baseline.

2 Corpus and Datasets

In this section, we elaborate the corpus of E-Manuals and the benchmark datasets we create. These datasets are used for pre-training and to test the performance of the QA algorithms.

2.1 Creating the corpus of E-Manuals used for pre-training

To perform pre-training, we create a large text corpus of E-Manuals by collecting and pre-processing (details in *suppl.*) text from 307, 957 pdf files downloaded from source[3]. All these pdf files serve as manuals for several categories of products and services, such as baby care, kitchen appliances, electronic goods, personal care, lawn, garden, etc. The variety prevents over-fitting to the E-Manuals of a specific product type. The details of the dataset have been summarized in Table 1. On plotting the word cloud (figure in *suppl.*) for the most frequently occurring terms, it is found that words that make sentences instructional and assertive e,g,. "avoid", "help", "handle", "leave", "print" are prominent .

Property	Value
No. of E-Manuals	307,957
No. of paragraphs	11,653,755
No. of sentences per paragraph	4.4
No. of words per sentence	20.2
Total number of words	∼1 Billion
Size of corpus (in GB)	∼11 GB

Table 1: Details of the E-Manual pre-training corpus used in terms of property-value pairs

Question Answering Dataset

We create datasets of different types which can act as benchmarks to test the performance of a E-Manual Question Answering algorithm under varied circumstances. We consider two most popular categories of consumer items, mobile and smart TV. For each of these categories, we take a representative E-manual and employ experts to curate questions covering all sections of these manuals. We also check what are the questions raised by smart TV users on online forums. Finally, we expand our domain to 40 devices of different categories and collect a small representative QA for them to check the versatility of the algorithm. For all our datasets, we decided to choose a single brand to have some sort of consistency across E-manuals, incidentally we chose 'Samsung' due to convenience (reasons detailed in *suppl.*). However, other popular brands could also be chosen, we believe that would not make much of a difference. Note, except for TechQA Dataset (Castelli et al., 2020) which

[3] www.manualsonline.com

is built from questions regarding general software based technical support and hardly contains any question pertaining to E-manual, to the best of our knowledge, no such similar dataset is available.

2.2 Question Answering Dataset from E-Manual

We have selected E-Manual of a Samsung S10 phone (s10)and a Samsung Smart TV/remote (Tv-) and created corresponding question-answer datasets with the help of expert annotators. Each section is carefully read by an annotator [4] and she has accordingly posed questions and marked certain sentences from the section as the answer. An E-Manual's sections were split among 3 annotators to reduce cognitive load. The annotators were non-native but fluent English speakers. Annotators also curated **paraphrased questions** where an already existing question is expressed differently, e.g., "How do I turn off sound notifications?" is paraphrased as "How can I mute all notification sound?". A crowdsource based quality assessment of the annotations is conducted (*detail in suppl*) and is found to be satisfactory. The stats of our datasets along with the TechQA Dataset (Castelli et al., 2020) are presented in Table 2.

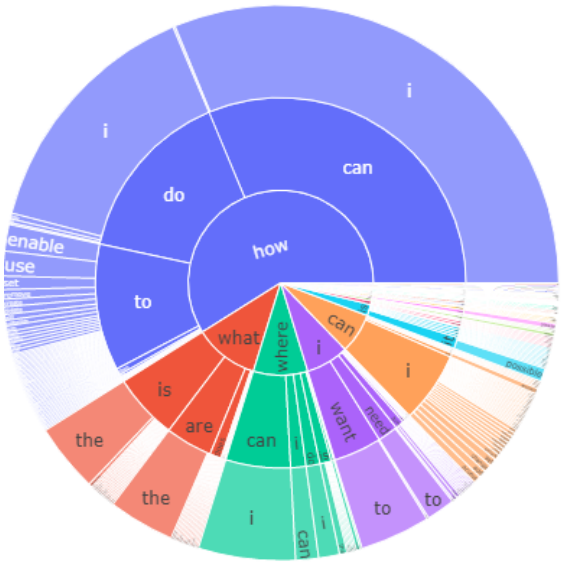


Figure 1: Distribution of questions covered in S10 QA Dataset w.r.t their first three tokens.

Most of the questions belong to one of these three categories - (a). about facts regarding device operations, which we refer to as "Factual". ('what', 'which', 'why', 'when' type questions) (b). on *how* to carry out a specific operation referred as "Procedural" ('how', 'can' type questions) (c) asking the location of a particular feature ('where' type questions). We show the distribution of questions w.r.t the first three tokens for Samsung S10 in Fig. 1. It shows that more than 50% of the questions are 'how' type questions ('how can', 'how to' etc.), while 'what', 'where' and 'can' type questions also have a significant percentage. There are also a few questions, which start with 'I want to', 'I need to', which start with the end user's desired functionality followed by a question ("I want to switch on Bluetooth. What should I do?").

2.3 Questions from the real consumers

The QA dataset of the Samsung Smart TV manual is used to sanitize a community-based question answering dataset described next. Questions are extracted from question answering forum (where well-formed answers are available) of the different Samsung Smart TV models sold on amazon. Annotators are asked to certify whether a question is answerable by solely using the E-Manual of the product. The dataset has a total of $3,000$ such questions, out of which $1,028$ are certified as answerable. Also, for each question, they were asked to select the most similar question from the manually annotated QA dataset created for Samsung Smart TV/Remote. This would provide paraphrases for the relevant Consumer Questions, and the Consumer Question-Annotated Question pairs so formed are referred to as the CQ-AQ Dataset. The CQ-AQ Dataset covers 312 of the annotated answers in the Smart TV/Remote QA Dataset, hence have the answer from the e-manual as the **ANNotated-Ground Truth (ANN-GT)**. The other Ground Truth for a CQ-AQ pair is the answer from the Amazon Community Question Answering (CQA) Forum corresponding to the CQ, which is the CQA-Ground Truth (CQA-GT). We thus create a dataset consisting of 1028 tuples, where each tuple consists of [CQ-AQ, ANN-GT, CQA-GT].

2.4 Questions spanning across several devices

In this step, we curate 10 generic Question-Answer pairs for 40 devices on Amazon [5]. We sample 10 questions from the S10 QA Dataset that would

[4] http://www.tika-data.com/

[5] 13 Samsung Galaxy Mobile Phones, 9 other Samsung Mobile Phones, 15 Samsung Tablets and 3 Samsung Smart Watches

Dataset	Domain	No. of QA pairs	%age of factual questions	%age of procedural questions	%age of questions asking feature location	%age of paraphrased questions	Avg Question Length	Avg. Answer Length	Answer Type
TechQA (Castelli et al., 2020)	Technical Support	1,400	22.75	32.64	0.88	0	52.5	45	Single Span, long answer
S10 QA	E-Manual	904	7.08	48.34	7.3	33.52	9.4	48.4	Multi Span, long answer
Smart TV/Remote QA	E-Manual	950	14.26	51.74	3.03	30.35	11	61.5	Multi Span, long answer
Smart TV/Remote Amazon Consumer Questions	User Forum	1,028	12.35	37.06	0.97	0	12.84	20.41	Multi Span, long answer

Table 2: Description of our datasets and the TechQA Dataset. The % showing various categories (including the paraphrase) does not sum upto to 100 as some questions cannot be classified into one of the three categories. The categories of the paraphrase is not shown as they roughly follow the similar distribution of the unique questions.

apply to a broad suite of devices. These 10 questions are sampled so that their corresponding annotated answers are from different sections of the E-Manual, and 1 is factual, 8 are procedural, and 1 is asking the location of a feature. These 10 questions are *listed in suppl.* We consider 40 devices of different types. For each device, for each of the 10 sample questions, the most relevant question is selected from the Amazon QA for that device using the CQ-AQ Paraphrase Detector *discussed in suppl.* The answer corresponding to each question from Amazon is taken as the ground truth answer. Thus, we have 10 question pairs and a corresponding set of 10 answers as the dataset for each of the 40 devices.

3 Methodology

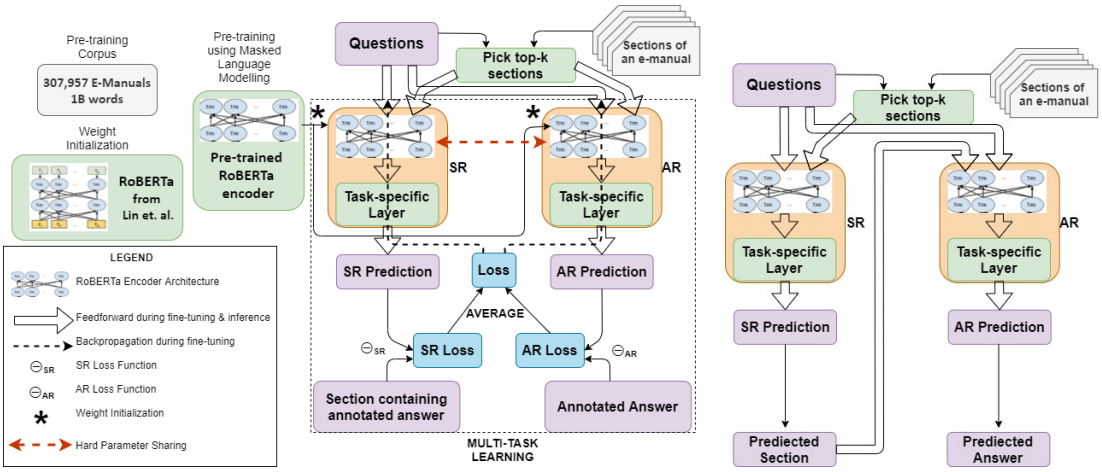
In this section, we describe each step from the pipeline of **EMQAP**. The pipeline consists of two major steps (a). **pre-training** the E-manual and (b). **multi-task learning** framework to select the answer. However, before employing multi-task learning, the first step is to reduce the pipeline's search space and provide it with only a few candidate sections for a question. We use an **unsupervised IR** method that accepts a question and all sections of the E-Manual as input and provides similarity scores for each question-section as output (*details in suppl.*) The flow of the entire EMQAP is depicted in Fig. 2. The steps are also presented as Algorithm in *suppl.*.

3.1 Pre-training on the E-Manuals corpus

A huge corpus of E-Manuals is used to pre-train the RoBERTa transformer using masked language modeling by masking 15% of the tokens in each input string to enhance the domain-specific knowledge of our language model. Note, the base "RoBERTa" transformer architecture is already initialized by weights obtained by pre-training it on Wikipedia, and BooksCorpus (Liu et al., 2019).

We apply the following two pre-training strategies to efficiently capture both the generic and domain-specific knowledge required to answer a question. (a). Using a learning rate that linearly decreases by a constant factor (LRD) from one layer to the next, with the outermost language modeling head layer having the maximum learning rate, as in Arumae et al. (2020). This enforces a constraint that outer layers adapt more to the E-Manual domain, while the inner layers' weights do not change much, thus restricting them to retain the knowledge of the generic domain primarily. (b). Using elastic weight consolidation (EWC) (Kirkpatrick et al., 2017; Arumae et al., 2020) to mitigate catastrophic forgetting while switching from the generic domain on which original "RoBERTa" was pre-trained to the domain of E-Manuals. A batch size of 64 is used. Since our corpus size (11GB) is quite small compared to the datasets used for pre-training in Liu et al. (2019), we use a smaller batch size than used in Liu et al. (2019). However, the number of tokens per sentence is 20.2, which ensures that a batch has a large number of tokens even with a smaller batch size. We pre-train for 1 epoch since the training loss reaches a plateau, and does not reduce further at the end of the epoch. More details and justification for choosing the above mentioned techniques are *detailed in suppl.*

We wanted to have a subjective analysis as to how pre-training helped the model learn better domain-specific context. We compared the model with off-the-shelf RoBERTa Model. Top 100 most frequent words (excluding stopwords and numbers) present in the first 100,000 lines of the EManuals Corpus are taken. For each word, top 5 neighbours (based on cosine distance) are calculated for each model. The word and its neighbours are much more contextually related (through manual analysis) in case of RoBERTa pretrained on E-Manuals, showing that, pre-training on E-Manuals enhances the context and meaning of domain-specific words. 10 such samples are shown in Table 3.



(a) EMQAP: Pre-training and Multi-Task Learning

(b) Inference of EMQAP

Figure 2: EMQAP: RoBERTa architecture is used for pre-training with E-manuals, and its weights are used to initialize the SR and AR models of the MTL framework. A question along with the top K relevant sections form inputs to the SR and AR modules of the MTL Framework during training, and an average of the AR and SR losses is backpropagated through the whole framework. **During inference**, once top-k sections are retrieved from the unsupervised IR, the SR module outputs the most relevant section for the question; the question along with this predicted section are sent as input to the AR module, which finally predicts the answer to the question.

Word	Top 5 nearest neighbours for RoBERTa	Top 5 nearest neighbours for RoBERTa pre-trained on E-Manuals
key	**button**, ip, must, field, note	**press**, note, **click**, **button**, parameter
address	support, phone, message, button, change	name, **server**, **message**, **network**, local
port	operation, enabled, must, unit, enable	**ports**, **ip**, **server**, **device**, unit
support	control, description, address, ports, settings	**information**, **service**, **call**, **3com**, **web**
switch	operation, **change**, enabled, unit, **button**	**ip**, **ethernet**, **protocol**, **remote**, **telephone**
enabled	**enable**, enter, ui, operation, guide	**connected**, **enable**, device, **configured**, setting
change	one, call, time, **switch**, click	enter, enable, **new**, set, access
click	change, call, check, view, time	**press**, **key**, **button**, enable, ip
button	**phone**, local, may, figure, **switch**	**click**, **key**, **remote**, displays, **router**
figure	button, **table**, may, local, unit	**data**, **example**, **see**, **line**, guide

Table 3: 5 nearest neighbors for domain specific words, where the words are represented as the output given by the last hidden layer of either RoBERTa from (Liu et al., 2019) or RoBERTa pre-trained on the corpus of E-Manuals, further compressed into a 3-D vector using PCA (F.R.S., 1901). For each word, most related neighbours are highlighted in **bold**

3.2 A Multi-Task Learning Approach for SR and AR

In our MTL framework, SR and AR models are sequential classification networks that consist of a RoBERTa encoder followed by a task-specific classification layer. The objective of the SR model is to retrieve the section which is most relevant to the question. The objective of the AR model is to retrieve the answer to the question from that section. For this, we use two settings - sentence-wise and token-wise classification.

Both SR and AR branches share the feature extraction layers of the "RoBERTa" architecture. It is well known that such a 'hard parameter sharing' approach (Caruana, 1993) greatly reduces the problem of overfitting. Each branch has a task-specific (here task refers to one of SR or AR) binary classification layer at the end, where the output is 2 dimensional for the SR as well as the sentence-wise AR, whereas, the output has a dimension of $(n_t \times 2)$ in case of the token-wise AR, where n_t represents the number of tokens in the input section.

Our architecture used has similarity with Nishida et al. (2018); however, ours is an improved shared transformer architecture with self-attention and skip connections (Vaswani et al., 2017), as compared to their shared Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) layers. Also, we predict non-contiguous sentences and non-contiguous spans, which makes the task difficult due to the need for detecting long-range dependencies, and thus improves the answer retrieval as compared to Nishida et al. (2018). The underlying domain-specific pre-training of RoBERTa provides the architecture the necessary boost to capture such difficult constraints.

Training: Given a question, we perform the following feed-forward approach for each section

retrieved by the unsupervised IR method. During sentence-wise classification, the AR model takes the question, and a sentence from the current section as input, and the SR model takes the question and the current section as input. Whereas, during token-wise classification, the AR and SR models both take the question and the current section as input. The targets are to set to 1 or 0 as per the relevance of the sentences/tokens. During back-propagation, the multi-task loss L_{MT} is the average of the loss for SR and AR (similar to Sun et al. (2020)).

4 Experiments and Results

To assess the efficiency of EMQAP, we first evaluate the performance of the unsupervised retrieval algorithm followed by the MTL Framework on the datasets specifically curated in Sections 2.2 – 2.4. The experimental results of unsupervised algorithm is *detailed in suppl.* We found that the proposed algorithm **TF-IDF + T5** performs the best.

4.1 Experimental Setup

We set the unsupervised IR method to **TF-IDF + T5**. Also, we take top $K = 10$ sections retrieved given a question as input to the supervised method, since one achieves almost 94% HIT when the top-10 retrieved sections are considered. The MTL network fine-tunes the pretrained model using the S10 dataset. The fine-tuning is done with a batch size of 32, and early stopping is applied using the validation loss. The Samsung S10 dataset, which consists of 904 question-answer pairs with 303 paraphrased question pairs is divided into three sets - 634 samples in the training set, 180 samples in the validation set, and 90 samples in the test set. The division ensures the paraphrased questions all fall in the same set. [The test datasets are a bit different in Sec. 4.5 and Sec. 4.6.]

4.2 Metrics

We use the following metrics for evaluation of the **MTL** framework. **(a). Exact Match** - Fraction of times the predicted answer and ground truth exactly match. **(b). ROUGE-L (Lin, 2004)** - F-measure metric designed for evaluation of translation and summarization. It is evaluated based on the longest common subsequence (LCS) between the actual answer and the answer predicted by a question-answering method. **(c). Sentence and Word Mover Similarity (Clark et al., 2019)** - In the case of the S+WMS metric, the GloVe word embeddings (Pennington et al., 2014) are weighted by the word frequencies, and the sentence embeddings (obtained by averaging the GloVe word Embeddings) are weighted by the sentence lengths, and a bag of words and sentence embeddings is created. To obtain the similarity value, a linear programming solution is used to measure the distance a predicted answer's embedding has to be moved to match the actual answer.

4.3 Evaluating MTL framework

Baselines: We compare EMQAP with other baselines such as

(A) *Method based on efficient passage retrieval*

Dense Passage Retrieval (DPR) (Karpukhin et al., 2020): A dual BERT (Devlin et al., 2019) encoder framework is used for retrieving relevant sections, and after retrieving the relevant sections, it assigns a passage selection score to each passage. Finally, a span selection method selects the span from the section with the highest score as the answer. We fine-tune the dual-encoder framework and the span selector on our dataset.

(B) *Methods with efficient answer retrieval*

Technical Answer Prediction (TAP) (Castelli et al., 2020): **TAP** uses a cascaded architecture, where a **document ranker** ranks the top documents (here, sections) according to an assigned score, and the section with the highest score is passed to a **span selector**, which predicts the answer span. This baseline is of significance, as it has been used for the TechQA Dataset, which is the closest to our dataset in terms of the domain.. Both the **document ranker** and the **span selector** are based on the **BERT-BASE-UNCASED** architecture, and we fine-tune both of these on S10 QA training dataset.

MultiSpan (Segal et al., 2020): This method solves Question Answering using a sequence tagger based on the RoBERTa (Liu et al., 2019) architecture (we use RoBERTa-BASE architecture, as opposed to RoBERTa-LARGE as mentioned in the paper). It predicts for each token whether it is part of the answer. For a question, the most relevant section is extracted using an IR method, and the sequence tagger is then fine-tuned using our QA Dataset. This method is of significance, as it predicts multiple spans as the answer, which matches the nature of our QA dataset.

Results: Table 4 enlists the exact match, ROUGE-

L precision, recall, F1 and S+WMS scores of these baselines, along with those of sentence-wise and token-wise classification version of EMQAP. **MultiSpan** has the highest ROUGE-L precision, and **EMQAP-S** is a close second. **TAP** is the best baseline when ROUGE-L F1 Scores and S+WMS scores are compared. However, **EMQAP-S** and **EMQAP-T** perform significantly better than **TAP**, both having p-values of approx. 0.029. EMQAP beats all baselines, when it comes to exact match (almost no algorithm could retrieve even a single exact ground truth), S+WMS, ROUGE-L recall and F1-Scores for the following reasons - (1) The **DPR** method, although having an efficient passage retrieval, cannot select multiple spans. (2) Although **TAP** performs well on TechQA Dataset, it performs inferior to our method, as it cannot handle multiple spans. However, it performs better than other baselines overall, as it can give a long span as an answer, by splitting a document/section into two inputs, and later concatenating the $< START >$ token representations (3) Although **MultiSpan** can extract multiple spans as answers from a section, answer spans present in our dataset have many tokens, which could not be handled by a Sequence Tagging Method, hence giving high ROUGE-L precision, but poor metrics otherwise. **DPR** and **MultiSpan** tend to predict very short answers, which can explain their low recall. We present examples of different question types and their predictions by the baselines along with ground truths in the *suppl.*

MODEL	EM	P	R	F1	S+WMS
DPR	0	0.646	0.174	0.256	0.021
TAP	0.133	0.448	0.466	0.426	0.284
MultiSpan	0	**0.938**	0.14	0.226	0.014
EMQAP-T	0.156	0.577	0.682	0.588	0.34
EMQAP-S	**0.311**	0.801	**0.541**	**0.604**	**0.354**

Table 4: Comparison of state-of-the-art models with EMQAP. (EMQAP-S and EMQAP-T are the Sentence-Wise and Token-Wise Classification variants, respectively)

4.4 Evaluating Pretraining techniques

The pretrained model can be trained with different learning rates and decay. Here we consider **(a). FT RB**: Fine-Tuning RoBERTa (Liu et al., 2019) **(b). SLR (Same Learning Rate)**: pre-train RoBERTa on E-Manuals with Learning Rate of 5×10^{-5} across all layers **(c). LRD (Learning Rate Decay)**: pre-train RoBERTa on E-Manuals with Learning Rate decaying linearly across layers by a factor of 2.6, the maximum learning rate being 5×10^{-4}. **(d). EWC**: pre-train RoBERTa on E-Manuals with Elastic Weight Consolidation (EWC) **(e). EWC+LRD**: Combination of **EWC** and **LRD**. The strategies c, d, and e have been discussed in detail in Section 3.1. Note as mentioned in Section 3.1 EMQAP uses **EWC+LRD**.

The efficacy of each of the pre-trained model can be evaluated from the performance in QA system. To solely concentrate on the pre-training performance, we consider a sequential model SQP (instead of MTL) where an SR system is followed by an AR system, and each system is trained separately. Both the SR and the AR architectures are the same as that of the SR and AR branches of the MTL framework described in Section 3.2.

Results: The results are shown in Table 5. Among the sentence-wise and the token-wise classification variants, the **SQP(EWC+LRD)** gives the best results considering exact match, ROUGE-L F1 and S+WMS scores, while the SQP(SLR) and the SQP(FT RB) variants perform the poorest among the lot, which is consistent with the results in Arumae et al. (2020). It only produces short answers, hence have a high precision but is poor on all other counts. Also important to note that each EWC and LRD contribute to the improvement in performance as performance of SQP with either EWC or LRD is inferior than when combined. Thus the result provides justification of using **EWC+LRD** for EMQAP.

Results: MTL over sequential learning: EMQAP using the **EWC+LRD** pre-training technique performs better than the best variant in all these three metric values compared to the respective sentence/token-wise classification regime. Overall, EMQAP performs better than best variant significantly with a p-value of 0.047. Also, the sentence-wise model gives a higher precision, while a token-wise model gives a higher recall. This could be attributed to the sentence-wise model, in general, giving a subset of the ground truth, while the token-wise model predicting more tokens than were in the ground truth. Another metric in which sentence-wise models perform better than Token-wise classification models is Exact Match, as the token-wise models tend to miss out on some tokens in each sentence of the predicted answer. We present examples of different question types and their predictions by the variants along with ground truths in the *suppl.*

	Sentence-Wise Classification					Token-Wise Classification				
MODEL	**EM**	**P**	**R**	**F1**	**S+WMS**	**EM**	**P**	**R**	**F1**	**S+WMS**
SQP(FT RB)	0.178	0.696	0.457	0.506	0.273	0.133*	**0.59**	0.602	0.566	0.335
SQP(SLR)	0.156	0.733	0.473	0.522	0.246	0.033	0.587*	0.668	0.579	0.302
SQP(LRD)	0.256	0.783	0.507	0.57	0.321	0.089	0.559	0.603	0.539	0.295
SQP(EWC)	0.233	0.763	0.511	0.552	0.285	0.1	0.554	0.634	0.575	0.314
SQP(EWC+LRD)	0.278*	0.791*	0.523*	0.592*	0.33*	0.133*	0.574	0.673*	0.583*	0.337*
EMQAP	**0.311**	**0.801**	**0.541**	**0.604**	**0.354**	**0.156**	0.577	**0.682**	**0.588**	**0.34**

Table 5: QA Evaluation on S10. "TF-IDF+T5" is applied by all the listed methods to select the top-10 relevant sections per question. EM stands for fraction of Exact Match. P(Precision), R(Recall) and F1 scores correspond to ROUGE-L (Lin, 2004). Best result for each metric is in **bold**, while the second best is marked with *

GT	EM	P	R	F1	S+WMS
AGT	0.304	0.778	0.522	0.582	0.332
CGT	0.049	0.362	0.297	0.306	0.278

Table 6: QA Evaluation on questions from CQA against corresponding answers from E-Manual of Samsung Smart TV as well as CQA. AGT is short for ANN-GT and CGT is short for CQA-GT ("TF-IDF+T5" is applied before all of the listed methods to select the top-10 relevant sections per question)

4.5 Evaluating Smart TV annotated on CQA Forums

We use the CQ-AQ Paraphrase dataset described in Section 2.3. The 1028 pairs of answerable questions and corresponding annotated answers from the manual (ANN-GT) and answers from CQA Forums (CQA-GT) are used to evaluate EMQAP.

Results : The results obtained are tabulated in Table 6. It is found that the results obtained on ANN-GT of Smart TV is inferior to that obtained on tested on S10 in Table 5. This happens because EMQAP is specifically fine-tuned on S10. However, we find that the performance deteriorates only a bit, pointing to the versatility of the fine-tuning.

It is found that the Exact Match and ROUGE-L F1-Scores are not as good for the ground truths of CQA-GT as compared to ANN-GT, which could be due to different kinds of n-grams present in CQA-GT and ANN-GT, as CQA-GT has a lot of personal opinions from users in addition to the actual solution to the problem being posed in the question, while, ANN-GT, being annotated from the E-Manual, is more impersonal and informative. However, the Mover Similarity Metrics for ANN-GT and CQA-GT are comparable which suggests that ANN-GT and CQA-GT are semantically similar. Hence, the Forum data can also act as a good ground truth, which we use in the next experiment.

4.6 Evaluation on several devices

EMQAP is evaluated on the set of 10 annotated questions for each device, the details of which are provided in Section 2.4. The averaged S+WMS Scores for the 4 categories (here, sentence-wise classification is used) are tabulated in Table 7. The mobile phones and tablets give similar results, as they have similar functionalities as S10, whereas smartwatches do not fair as well, as their functionalities differ from that of S10. SQP(EWC+LRD) performance is inferior reiterating the importance of MTL.

Sentence Wise Classification	Samsung Galaxy Mobile Phones	Other Samsung Mobile Phones	Samsung Tablets	Samsung Smart Watches
MTL (EMQAP)	**0.282**	**0.275**	**0.265**	**0.213**
SQP(EWC+LRD)	0.264	0.261	0.255	0.206

Table 7: Average S+WMS scores on CQA Forum for 4 categories across 40 devices for EMQAP and variants, fine-tuned on S10 dataset. Best result for each category is in **bold**, while the second best is marked with *

5 Conclusion

In this paper, we worked on a far less studied problem of question answering from E-Manuals. In order to work the subject, a pre-condition was to create benchmark datasets which we painstakingly developed. We created a large corpus from E-manuals which was used in pre-training a RoBERTa architecture. This in turn helped in developing a domain-specific natural language understanding; the fruits of which can be observed in the huge improvement in performance over competing baselines. We believe that the E-manuals specific QA dataset is extensive and well-rounded and will help the community in various ways.

Acknowledgements

We would like to thank the annotators who made the curation of the datasets possible. Also, special

thanks to Manav Kapadnis, an Undergraduate Student of Indian Institute of Technology Kharagpur, for his contribution towards the implementation of certain baselines. This work is supported in part by the Federal Ministry of Education and Research (BMBF), Germany under the project LeibnizKI-Labor (grant no. 01DD20003). This work is also supported in part by Confederation of Indian Industry (CII) and the Science & Engineering Research Board Department of Science & Technology Government of India (SERB) through the Prime Minister's Research Fellowship scheme. Finally, we acknowledge the funding received from Samsung Research Institute, Delhi for the work.

References

Link to the samsung s10 smartphone e-manual. https://downloadcenter.samsung.com/content/PM/202001/20200128065515543/EB/UNL_G970U_G973U_G975U_EN_FINAL_200110/start_here.html.

Link to the samsung smart tv/remote e-manual. https://www.manualslib.com/manual/1368844/Samsung-Smart-Remote.html#manual.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Kristjan Arumae, Qing Sun, and Parminder Bhatia. 2020. An empirical investigation towards efficient multi-domain language model pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4854–4864. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3613–3618. Association for Computational Linguistics.

Richard Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Morgan Kaufmann.

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, J. Scott McCarley, Mike McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John F. Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avirup Sil, Rosario Uceda-Sosa, Todd Ward, and Rong Zhang. 2020. The techqa dataset. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1269–1278. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Elizabeth Clark, Asli Celikyilmaz, and Noah A Smith. 2019. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2748–2760.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Karl Pearson F.R.S. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

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

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. 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. *arXiv preprint arXiv:1907.11692*.

Kyosuke Nishida, Itsumi Saito, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2018. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 647–656.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

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, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Elad Segal, Avia Efrat, Mor Shoham, Amir Globerson, and Jonathan Berant. 2020. A simple and effective model for answering multi-span questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3074–3080, Online. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. In *AAAI*, pages 8968–8975.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Ming Zhu, Aman Ahuja, Da-Cheng Juan, Wei Wei, and Chandan K. Reddy. 2020. Question answering with long multiple-span answers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3840–3849, Online. Association for Computational Linguistics.