13 - DrFAQ: Benchmarking and Analysis of Language Model Transfer Learning for a Plug-and-Play Question Answering Chatbot

Authors: New Jun Jie, Elysia Tan Ziyi, Chua Xinhui, Sarah, Chik Cheng Yao (School of Computing)
Email: e0389098@u.nus.edu, e0036110@u.nus.edu, e0035831@u.nus.edu, e0174850@u.nus.edu

Abstract

With the rise of deep Natural Language Processing (NLP), companies race to streamline processes by employing question answering (QA) to automate interaction services via chatbots. DrFAQ is an open-source QA chatbot architecture, to which we extend by improving and analysing the NLP QA procedure within. Our contributions include benchmarking 6 language models (LMs) BERT, DistilBERT, RoBERTa, DistilRoBERTa, ALBERT and MobileBERT by their zero-shot and fine-tuned performance, and analysing QA capabilities by question categories, on 3 existing QA datasets (SQuAD, CoQA and QuAC) and 11 QA datasets generated from company information found online. Our experiments empirically show that RoBERTa performs best for large and clean QA datasets while MobileBERT performs best for small and unclean generated QA datasets. We contribute code for the transfer learning procedure, dataset generation and question classification.

1 Introduction

Industry trends show that companies are searching for ways to improve customer service experience while reducing costs. A leading solution is to automate the service process using chatbots. For instance, DBS uses a question answering (QA) chatbot called Jim to address job applicants' questions regarding the organization and the role they are applying for. However, the majority of current chatbots, like Jim, are rule-based with little machine learning or deep learning involved (DBS, 2018). Consequently, most chatbots currently available in the industry are highly specific to a single use case, and the creation of chatbots is highly manual and laborious.

Nonetheless, in recent years, the success of deep Natural Language Processing (NLP) has sparked interest in applying deep NLP to QA chatbots to make them more flexible and suitable for a wider range of use cases. For example, over 70 Singapore government agency websites use a single NLP QA chatbot, Jamie, which is able to tailor her answers to the specific website (GovTech, n.d.).

Thus, a natural tendency is to investigate the practicality of creating a plug-and-play QA chatbot, comprising a language model (LM) that can be generally fine-tuned to any organization's text corpora (i.e. uses the same fine-tuning methods regardless of text corpora). This can help to automate the manual process of creating customised QA chatbots, improving manpower efficiency and reducing costs.

DrFAQ is an existing open-source project by a team member that implements a 4-step QA methodology: 1) FAQ matching, 2) NLP QA, 3) Search and 4) Human intervention. However, DrFAQ has some limitations. It is based on an arbitrarily selected BERT LM pre-trained on SQuAD, which is not fine-tuned to any target QA datasets, and was not subject to any evaluation studies to investigate its limitations. In particular, it is unknown whether the pre-trained BERT LM can be adapted to the contexts of different companies.

Therefore, our project aims to extend DrFAQ by improving and analysing the NLP QA procedure within, through our following contributions:

1. We benchmark 6 LMs (BERT, DistilBERT, RoBERTa, DistilRoBERTa, ALBERT and MobileBERT) instead of just one;
2. We generated 11 QA datasets from company information found from websites and Wikipedia pages, which we call "corpora-of-interest" or CoI, alongside the 3 QA research datasets commonly used in the QA literature (SQuAD 1.1, CoQA and QuAC) to simulate the real world use case.
3. We perform zero-shot QA using the LMs on the 14 datasets to evaluate their zero-shot performance based on the F1 and Exact Match (EM) metrics.
4. We further fine-tune the LMs (beyond the initial fine-tuning on SQuAD) on all datasets except SQuAD to evaluate their fine-tuned performance based on F1 and EM, and assess the extent of catastrophic forgetting on the SQuAD dataset.
5. We analyse the degree of improvement of LMs and their limitations on all datasets by question categories, classified using a rule-based heuristic.
6. We contribute code for the transfer learning procedure, QA CoI dataset generation and question type classification.

In particular, we investigate the following 2 research questions (RQ):

1. Which LM, after fine-tuning on different QA datasets, adapts best to new ones?
2. What errors and limitations in question answering do these LMs have, and why?

By understanding the qualities and capabilities of each model, companies will be able to make a more informed decision with regards to the LM that will best suit their own use case.

2 Related Work

With the efforts of the open-source and NLP community, there exists thousands of readily-available pre-trained and fine-tuned LMs on different datasets and using various combinations of training hyperparameters, many of which can easily be used as the LM base for an extractive QA system.

Adoma et al. (2020) compared several LMs - BERT, RoBERTa, DistilBERT and XLNet - and their relative efficacy in recognising emotions from text. When the same hyperparameters were used, RoBERTa performed the best with 0.743 accuracy while DistilBERT performed the worst with 0.601 accuracy, on the International Survey Emotion Antecedents and Reactions dataset. As can be seen, with the many available LMs to choose from, there is likely a model that performs better in certain contexts. Our project is similar but differs from Adoma et al. (2020) in that we are comparing the efficacy of LMs in terms of QA capabilities.

Pre-trained LMs are commonly used in QA systems by fine-tuning on the target QA dataset. Min et al. (2017) studied transfer learning, in the context of QA, with BiDAF by using SQuAD as the source dataset and WikiQA and SemEval 2016 as the targets. They found that directly training the LM on the target dataset resulted in poorer performance than when it was first pre-trained on SQuAD. This provides support for the methodology adopted in this project where LMs are first trained on SQuAD before being fine-tuned on the target dataset. Our project differs in the choice of LMs and datasets experimented with. Furthermore, we conduct in-depth analysis to understand the QA capabilities of the LMs.

TANDA (Garg et al. 2019) is a recent work that we take reference from. TANDA (Transfer-and-Adapt) is a two-step fine-tuning process where the LM is initially fine-tuned on a general, large and high-quality QA dataset such as SQuAD to develop QA capabilities, and then further fine-tuned on the target QA dataset to adapt the LM to the target domain. The main advantage of TANDA is its ability to adapt the LM to small target QA datasets because of the initial fine-tuning on a general QA dataset. Therefore, TANDA is a compelling method to be used to employ DrFAQ's goals, where companies may not have a large dataset or text corpora. Our project differs from Garg et al. (2019) in terms of the analysis done, we focus on understanding the QA capabilities of LMs.

3 Language Model

Since one of the RQs of this project was to identify the LM that is best able to adapt to new datasets, six LMs were experimented with. Given that BERT was a landmark LM and a well-studied baseline, we chose to focus our study on BERT and its related LMs - RoBERTa, DistilRoBERTa, DistilBERT, MobileBERT and ALBERT.

BERT	RoBERTa	DistilRoBERTa	Distil BERT	Mobile BERT	ALBERT
110	125	82	66	25.3	12

Table 1: No. of Parameters in LMs (millions)

The BERT model (bert-base-uncased) contains 12 layers of transformer blocks each with 12 attention heads and 768 hidden layers resulting in a total of 110 million parameters (Devlin et al., 2019).

The RoBERTa model (roberta-base) used has the same architecture as BERT, except that it has a total of 125 million parameters instead of 110 million (Liu et al., 2019).

The DistilRoBERTa model (distilroberta-base) used has the same architecture as RoBERTa, except that it has 6 layers of transformer blocks instead of 12 resulting in a total of 82 million parameters (HuggingFace, n.d.).

The DistilBERT model (distilbert-base-uncased) used has a similar architecture as BERT, except that it has 6 layers of transformer blocks, half that of BERT, resulting in a total of 66 million parameters (Sanh et al., 2019).

The same distillation process was used in both DistilBERT and DistilRoBERTa. For DistilBERT, the model was initialised using alternate layers from BERT. It was then trained on a concatenation of English Wikipedia and Toronto Book Corpus using a training loss which is a linear combination of the distillation loss over the soft target probabilities of the teacher (original BERT), masked language modelling loss and cosine embedding loss (Sanh et al., 2019).

The MobileBERT model (mobilebert-uncased) used has a slightly different architecture from BERT. It has 24 layers of transformer blocks, twice that of BERT. Each block has 4 attention heads and 128 hidden layers resulting in a total of 25.3 million parameters (Sun et al., 2020).

Similar to BERT, MobileBERT was trained on a concatenation of English Wikipedia and Toronto Book Corpus using a distillation loss which is a linear combination of the original masked language modelling (MLM) loss, next sentence prediction (NSP) loss and a new MLM Knowledge Distillation loss. Additionally, the model was trained using Progressive Knowledge Transfer where knowledge is transferred from the teacher (inverted-bottleneck BERT) to the student (MobileBERT) layer by layer (Sun et al., 2020).

The ALBERT model (albert-base-v2) used has the same architecture as BERT, except that it only has 12 million parameters as compared to 110 million in BERT. It employs cross-layer parameter sharing for all parameters (Lan et al., 2019) which contributed to the lower number of parameters despite the same architecture.

4 Data

Since different companies have different use cases, some companies may use DrFAQ for answering new employees' queries during onboarding and others may use it for answering customers' frequently-asked questions (hence 'FAQ') on their customer-facing company website. As such, RQ1 investigates the transferability of LMs to QA datasets that may come from customer-facing websites that are phrased to sound attractive to customers but less informative, or from factually-phrased corpora-of-interest (CoI) such as Standard Operating Procedure (SOP) documents.

Thus, aside from the research QA datasets, SQuAD 1.1, CoQA and QuAC, we generate 11 QA datasets from the CoI (Tables 2 and 3). From the corpora, passages that are in full sentences are extracted (excluding headers), questions and answers are then generated sentence-by-sentence using a multi-task QA-QG library (Patil, 2021).

Website	Samples
ByteDance	78
Discord	67
Reddit	62
Patsnap	69
Ripple	72
CS4248	368

Table 2: No. of Samples in Website-Generated Datasets

Wikipedia	Samples
ByteDance	120
Discord	207
Reddit	568
Spotify	473
Strava	52

Table 3: No. of Samples in Wikipedia-Generated Datasets

Websites and Wikipedia pages of the industry companies were chosen so as to obtain a mix of differently-phrased text corpora. Company websites are meant for customers to view and do not contain much factual information, while Wikipedia pages act as a knowledge base of company information, so datasets generated from websites tend to be smaller and messier while datasets generated from Wikipedia pages tend to be larger and cleaner (see examples in Appendix A), with the exception of the CS4248 website, which serves as a source of course information for class students.

5 Software and Hardware

All work was done in Python 3.7, using the HuggingFace library for access to QA datasets, pre-trained LMs and the fine-tuning procedure. NLTK was used for data pre-processing prior to the open-source QA-QG library for dataset generation from company information (Patil, 2021). Matplotlib was used for data visualisation. All experiments were run using the GPUs available on Google Colab (Tesla K80, P100) and NUS Compute Clusters (Tesla V100).

6 Methodology

6.1 Zero-Shot Experiment

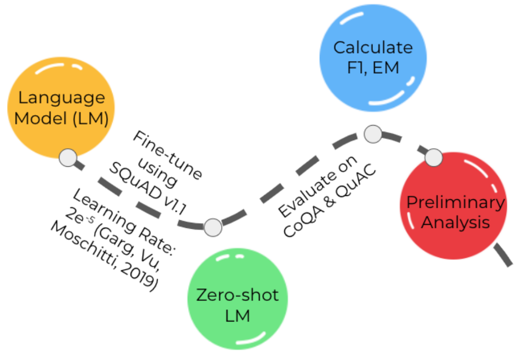

Fig 1: Procedure for Zero-Shot Experiments

The first experiment evaluates the zero-shot performance of LMs on target QA datasets. LMs are initially fine-tuned on SQuAD 1.1 using a linearly declining learning rate of $2e^{-5}$ (Garg et al. 2019) with a weight decay rate of 0.01. For all models, a train and evaluation batch size of 16 was used and the number of training epochs was set to 3. The maximum length of a question-answer pair was set to 384 and the maximum overlapping length of two contexts, when splitting was required, was set to 128 tokens. SQuAD was arbitrarily chosen as the initial fine-tuning dataset as it is one of the most well-studied QA dataset in the literature, and we leave alternative choices of initial dataset for future work. The QA models are then evaluated on the 3 QA datasets and generated datasets for their zero-shot performance.

6.2 Fine-Tuning Experiment

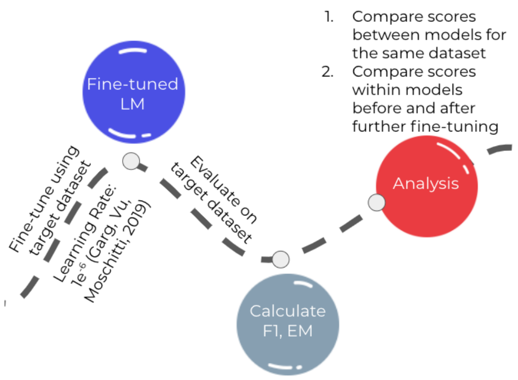

Fig 2: Procedure for Fine-Tuning Experiments

The second experiment evaluates the fine-tuned performance of LMs on target QA datasets. The LMs initially fine-tuned on SQuAD were further fine-tuned on the target QA datasets using the same fine-tuning procedure but with a smaller, linearly declining learning rate of $1e^{-6}$ as recommended by Garg et al. (2019). The further fine-tuned QA models were then evaluated on the 2 target QA datasets and 11 generated datasets for their performance.

One limitation of further fine-tuning is catastrophic forgetting, where capacities on the initial fine-tuned dataset is reduced after further fine-tuning on the target (Goodfellow, 2013). Recognising this limitation, we evaluate the further fine-tuned QA models on SQuAD to investigate the severity of catastrophic forgetting.

6.3 Model Evaluation

The languages are evaluated using the F1 and Exact Match (EM) scores (Rajpukar 2017).

$$F1\ Score = \frac{2 \times (Precision \times Recall)}{Precision + Recall}$$

The F1 score is computed for each question-answer pair, with precision and recall computed on a word-by-word basis:

$$Precision = \frac{Number\ of\ words\ in\ predicted\ answer\ in\ the\ same\ position\ as\ correct\ answer}{Length\ of\ the\ predicted\ answer}$$

$$Recall = \frac{Number\ of\ words\ in\ predicted\ answer\ in\ the\ same\ position\ as\ correct\ answer}{Length\ of\ the\ correct\ answer}$$

The F1 score of a LM evaluated on a dataset is therefore the average F1 score over all samples in the dataset.

$$Exact\ Match = \frac{Num\ of\ questions\ with\ predicted\ answer\ exactly\ matching\ correct\ answer}{Total\ number\ of\ questions}$$

The LM that scores higher on both F1 and EM is the superior model.

6.4 Analysis

To identify which LM adapts best to a new dataset (RQ1), F1 and EM scores of fine-tuned models are compared on each dataset and on average. The scores are also compared within models for zero-shot and fine-tuned QA performance to understand to what extent fine-tuning helped improve QA capabilities of each LM. The improvement from fine-tuning can be computed by the difference in performance pre- and post-fine-tuning. Further, it is unclear whether fine-tuning on a target, especially if it is relatively small and unclean, necessarily improves or might even worsen QA performance.

To understand the errors and limitations of various LMs in QA (RQ2), error analysis is conducted by categorising questions into different question types. Questions in each validation dataset are classified into their respective question types, namely *who, what, when, where, which, why* and *how*, using a rule-based heuristic of keywords and simple parsed syntax (Biswal et al. 2014), and *others* as a catch-all category. Some question types were further split into more granular categories, e.g. definition, descriptive or factoid questions, but given the very small sample size of the granular categories, the more general categories were used instead. We hypothesise that some question categories such as *why* and *how* may be more difficult than others like *who* and *what*. Analysing the relative performance of each question category therefore gives insight into the relative strengths and limitations of each LM.

LMs are fine-tuned on the full training dataset (inclusive of all question categories), then evaluated on the validation split segregated by question categories.

7 Experimental Results

7.1 Zero-Shot and Fine-Tuning Experiments on QA Datasets

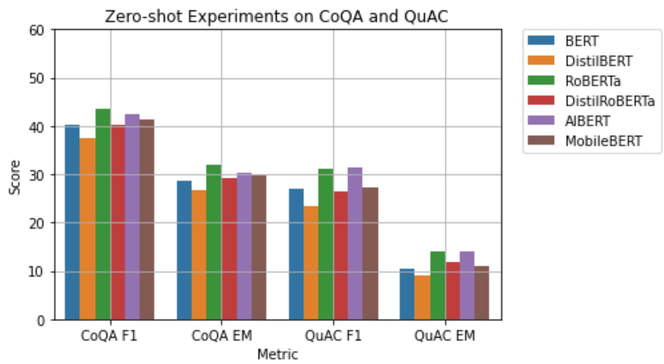

Fig 3: F1 and EM of Zero-Shot QA on CoQA and QuAC

As shown in Fig. X, RoBERTa scored the highest F1 and EM scores for zero-shot QA (CoQA F1: 43.6, EM: 32.0) (QuAC F1: 31.1, EM: 14.1). This empirical finding on zero-shot performance is consistent with the literature as RoBERTa is the largest LM with the most trainable parameters (Staliunaite et al. 2020).

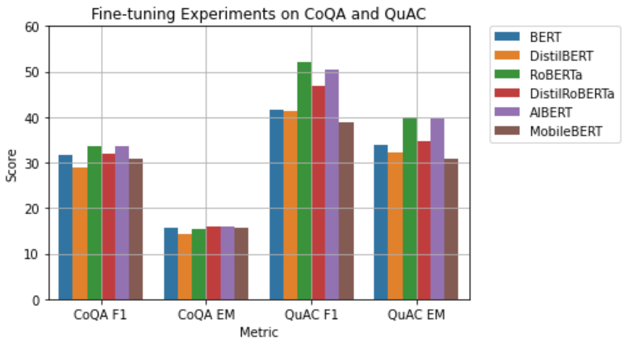

Fig 4: F1 and EM of Fine-Tuned LMs on CoQA and QuAC

As shown in Fig 4, RoBERTa also scored the highest F1 and EM on QuAC after fine-tuning (F1: 52.2, EM: 39.8), as expected of the largest LM. However, ALBERT scored the best F1 on CoQA after fine-tuning (F1: 33.6) and DistilRoBERTa scored the best EM (EM: 16.0). It is important to note that F1 and EM scores on CoQA across all LMs dropped after fine-tuning, with worst being MobileBERT by F1 (F1: -10.7) and RoBERTa by EM (-16.5), which we explore further in section 9. RoBERTa improved the most on QuAC by absolute value after fine-tuning (F1: +21.0, EM: +25.7).

7.2 Catastrophic Forgetting on SQuAD

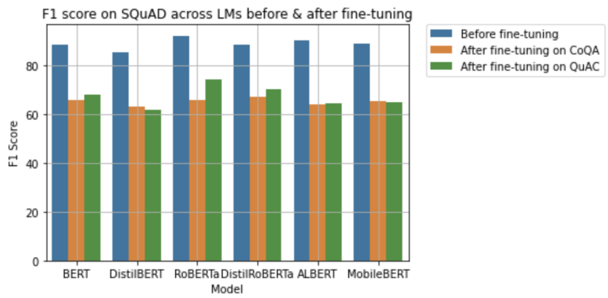

Fig 5: F1 scores Before and After Fine-tuning on CoQA and QuAC

As shown in Fig 5, all LMs suffered catastrophic forgetting on SQuAD. ALBERT suffered the worst catastrophic forgetting after fine-tuning on both CoQA and QuAC (CoQA F1: -26.3, EM: -43.7) (QuAC F1:-25.7, EM:-45.5). With the exception of EM on CoQA, DistilRoBERTa suffered the least from catastrophic forgetting (CoQA F1: -21.085) (QuAC F1: -18.080, EM: -32.439).

7.3 Zero-Shot and Fine-Tuning Experiments on Generated Datasets

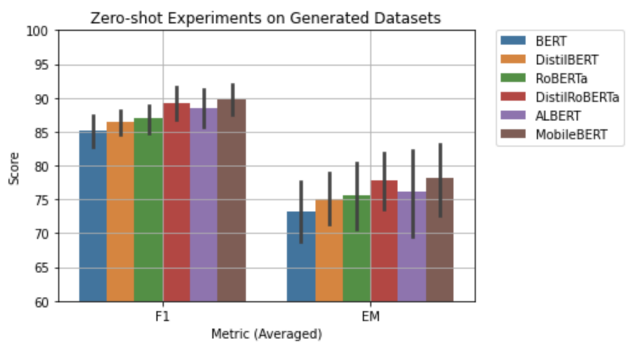

Fig. 6: Average F1 and EM of Zero-shot LMs on Generated Datasets

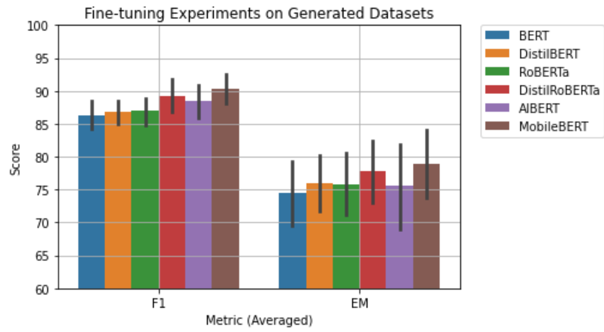

Fig 7: Average F1 and EM of Fine-Tuned LMs on Generated Datasets

As shown in Fig 6 and Fig 7, MobileBERT scored the highest average F1 and EM on zero-shot QA (F1: 89.5, EM: 79.5) across the generated datasets, and also the highest average F1 and EM after fine-tuning (F1: 90.1, EM: 80.2). RoBERTa and DistilRoBERTa negligibly improved on average (RoBERTa F1: 0.0, EM: +0.1) (DistilRoBERTa F1: +0.0, EM: 0.0) after fine-tuning while ALBERT worsened marginally (F1: -0.1, EM: -0.5).

8 Error Analysis

8.1 Error Analysis on Fine-Tuned QA by Question Categories

As can be seen from section 7.1, RoBERTa performed the best on the 2 research QA datasets and from section 7.3, MobileBERT performed the best on the generated datasets. Thus, in this section, for simplicity sake, the focus will be on RoBERTa for the research QA datasets and MobileBERT for generated datasets.

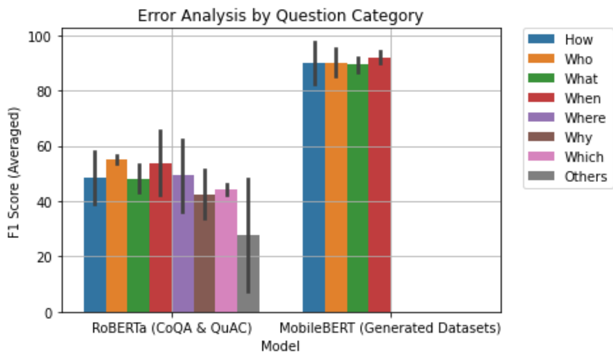

Fig 8: Average F1 of RoBERTa on QA Datasets and MobileBERT on Generated Datasets, by Question Category

From Fig 8, RoBERTa scored highest on 'Who' (F1: 55.2) and 'When' (F1: 53.9) questions and worst on 'Why' (F1: 42.6) and 'Others' (F1: 27.4) questions on the research QA datasets. MobileBERT scored best on 'When' (F1: 91.8) questions across the generated datasets. Across other types of questions, it is unclear which model is the best due to a small sample size or a lack of pattern.

8.2 Error Analysis on Catastrophic Forgetting

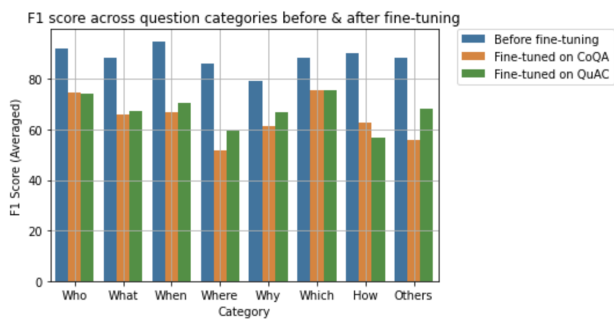

Fig 9: F1 Scores of LMs Before and After Fine-tuning on QA Datasets

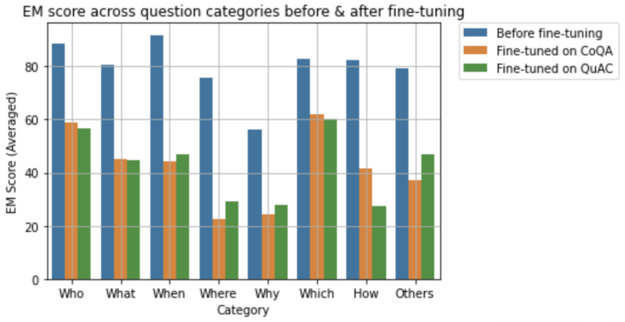

Fig 10: EM Scores of LMs Before and After Fine-tuning on QA Datasets

Fig 9 and Fig 10 show that the extent of catastrophic forgetting of SQuAD varies across question categories. When fine-tuned on CoQA, 'Where' questions suffered the biggest deterioration (F1: -34.3, EM: -53.5), followed by 'Others' and 'When' questions. When fine-tuned on QuAC, 'How' questions suffered the most forgetting (F1: -33.4, EM: -54.6), followed by 'Where' questions.

9 Discussion

In section 7.1, as shown in Fig 4, fine-tuning on CoQA resulted in an unexpected deterioration in QA performance. We conjecture the reason for the deterioration being that most questions in CoQA required conversation history to answer, due to the large frequency of coreferences in the questions (Yaskar et al. 2018). Examples of CoQA questions with coreferences can be referenced in Appendix A. This suggests some limitations on the generalisability of a LM to new datasets. Therefore, potential improvements can be made towards accounting for different types of datasets, such as dialogue-based QA datasets, for future work.

In section 6.1, we showed that on QuAC, RoBERTa performed best and DistilBERT performed worst. This is empirically consistent with Adoma et al. (2020)'s finding that RoBERTa performed best and DistilBERT performs worst on emotion recognition, albeit on a different task. While fine-tuned performance is evaluated on different tasks and domains, this suggests that the performance of RoBERTa > BERT > DistilBERT may hold for natural language tasks in general (Staliunaite et al. 2020, Cheang et al. 2020).

In section 6.3, we showed that MobileBERT performed best on generated datasets but not RoBERTa, even though RoBERTa performed best at CoQA and QuAC. We conjecture that MobileBERT performed better at generated datasets because of a smaller sample size and generally unclean dataset, and since MobileBERT is a smaller model so is less likely than RoBERTa to overfit to the generated datasets. Nonetheless, ALBERT is a smaller model yet performs worse than MobileBERT. We rule out the explanation that the cross-layer parameter-sharing architecture employed by ALBERT worsening the degree of catastrophic forgetting as ALBERT performed better than MobileBERT after fine-tuning on CoQA and QuAC. We acknowledge that due to many variables, such as architectural and parameter count differences, the reason for the superior performance of MobileBERT is difficult to identify, and we leave this for future work.

In section 7.1, we showed that 'Who' and 'When' questions were least difficult while 'Why' questions were the most difficult. This finding is not surprising given that 'Who' and 'When' questions are largely factual and thus easily referenced and extracted, while 'Why' questions usually require inference of entailment, especially when the passage does not phrase the answer explicitly. Nevertheless, it is important to note that, in the context of DrFAQ, the inability of the LM to answer 'Why' questions well is not a significant limitation of the LM as FAQs are generally factual in nature.

In sections 7.2 and 8.2, we showed that all LMs suffered from catastrophic forgetting after further fine-tuning on the target dataset. Catastrophic forgetting is a widely acknowledged problem of further fine-tuning and many researchers have proposed possible mitigations, such as by Chen et al. (2020) and Xu et al. (2020). Further, Rongali et al. (2020) showed that LMs which suffered less from catastrophic forgetting were more robust for the downstream task. As such, it will be important for companies to take this into account when training their own QA chatbots.

Overall, our benchmarking experiments and error analysis presents a few actionable insights for language model transfer learning. For a large and clean QA dataset, RoBERTa is the best LM. For a small and relatively unclean generated QA dataset, MobileBERT is the best LM. RoBERTa excels in 'Who' and 'When' questions while MobileBERT excels in 'When' questions. Full experimental results can be found in Appendix B.

10 Conclusion

In conclusion, from benchmarking and analysing language model transfer learning of BERT, DistilBERT, RoBERTa, DistilRoBERTa, ALBERT and MobileBERT, across 3 existing QA datasets and 11 QA datasets generated from company information found online, we empirically show that RoBERTa performs best for large and clean QA datasets while MobileBERT performs best for small and unclean generated QA datasets, and that 'Who' and 'When' questions are the least difficult while 'Why' and 'Others' questions are the most difficult. With these insights from our project, DrFAQ will be more robustly generalisable to different companies' use cases, depending on their size, quality and distribution of datasets.

References

[1] Abu-Mostafa, Yaser M., Magdon-Ismail, Malik and Lin, Hsuan-Tien. (2012) *Learning From Data*, AMLBook.

[2] Adoma, A. F., Henry, N. M., & Chen, W. (2020, December). Comparative Analyses of Bert, Roberta, Distilbert, and Xlnet for Text-Based Emotion Recognition. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)* (pp. 117-121). IEEE.

[3] Bishop, Christopher M. (2006) Pattern Recognition and Machine Learning. Springer.

[4] ByteDance. (2021, 9 April). In *Wikipedia.* https://en.wikipedia.org/w/index.php?title=ByteDance&oldid=1016864444.

[5] Cheang, B., Wei, B., Kogan, D., Qiu, H., & Ahmed, M. (2020). Language Representation Models for Fine-Grained Sentiment Classification. *arXiv preprint arXiv:2005.13619*.

[6] Chen, S., Hou, Y., Cui, Y., Che, W., Liu, T., & Yu, X. (2020). Recall and learn: Fine-tuning deep pretrained language models with less forgetting. arXiv preprint arXiv:2004.12651.

[7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[8] DBS. (2018). *DBS introduces Jim, Southeast Asia's first virtual bank recruiter*. DBS. https://www.dbs.com/newsroom/DBS_introduces_Jim_Southeast_Asias_first_virtual_bank_recruiter

[9] Discord. (2021, 9 April). In *Wikipedia.* https://en.wikipedia.org/w/index.php?title=Discord_(software)&oldid=1016844476.

[10] Garg, S., Vu, T., & Moschitti, A. (2020, April). Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 05, pp. 7780-7788).
[11] Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., & Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.
[12] GovTech. (n.d.). *'Ask Jamie' Virtual Assistant.* GovTech. https://www.tech.gov.sg/products-and-services/ask-jamie/
[13] HuggingFace. (n.d.). *distilroberta-base*. Hugging Face. https://huggingface.co/distilroberta-base.
[14] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.
[15] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
[16] Min, S., Seo, M., & Hajishirzi, H. (2017). Question answering through transfer learning from large fine-grained supervision data. *arXiv preprint arXiv:1702.02171*.
[17] Patil, S. (2021, February 17). *Question Generation using transformers.* GitHub. https://github.com/patil-suraj/question_generation
[18] Rajpurkar, P. (2017, April 3). *The Stanford Question Answering Dataset: Background, Challenges, Progress*. GitHub. https://rajpurkar.github.io/mlx/qa-and-squad/.
[19] Reddit. (2021, 11 April). In *Wikipedia.* https://en.wikipedia.org/w/index.php?title=Reddit&oldid=1017231764.
[20] Rongali, S., Jagannatha, A., Rawat, B. P. S., & Yu, H. (2020). Continual Domain-Tuning for Pretrained Language Models. *arXiv preprint arXiv:2004.02288*.
[21] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
[22] Spotify. (2021, 10 April). In *Wikipedia.* https://en.wikipedia.org/w/index.php?title=Spotify&oldid=1016971500.
[23] Staliūnaitė, I., & Iacobacci, I. (2020). Compositional and Lexical Semantics in RoBERTa, BERT and DistilBERT: A Case Study on CoQA. *arXiv preprint arXiv:2009.08257*.
[24] Strava. (2021, 9 April). In *Wikipedia.* https://en.wikipedia.org/w/index.php?title=Strava&oldid=1016807728.
[25] Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., & Zhou, D. (2020). Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*.
[26] Xu, Y., Zhong, X., Yepes, A. J. J., & Lau, J. H. (2020, July). Forget me not: Reducing catastrophic forgetting for domain adaptation in reading comprehension. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

Appendix A

Examples of Questions and Answers in Datasets:

Dataset	Questions	Answers
SQuAD 1.1	1. Who became president in 2013? 2. When were the free elections held?	1. Michel Djotodia 2. October 1992
CoQA	1. What colour was Cotton? 2. Where did she live?	1. White 2. In a barn
QuAC	1. What happened in 1983? 2. Did she have any other children?	1. In May 1983, she married Nikos Karvelas 2. CANNOT ANSWER
ByteDance (Website)	1. Who is the founder and CEO of ByteDance? 2. How many languages is Helo available in?	1. Yiming Zhang 2. 15
Discord (Website)	1. What is the name of the conversation? 2. What do people create for gaming, yoga classes, comedy fan clubs?	1. Organized Conversation. 2. Discord servers
Patsnap (Website)	1. What do we invest in? 2. What do our platform and collaboration tools seamlessly integrate into?	1. AI and innovation 2. integrate into your workflow
Reddit (Website)	1. What does one up or down vote mean? 2. What is the best way to gain karma?	1. +1 or -1 karma 2. submit posts that other people find valuable and interesting
Ripple (Website)	1. What is an alternative to pre-funding? 2. What page provides job opportunities?	1. On-Demand Liquidity 2. Careers
CS4248 (Website)	1. What rule states that you are free to meet with fellow students and discuss assignments with them? 2. What libraries will we use?	1. Pokemon Go Rule 2. SciKitLearn and PyTorch
ByteDance (Wikipedia)	1. Where did ByteDance release Resso? 2. Who is the Chinese-specific counterpart to TikTok?	1. India and Indonesia 2. Douyin
Discord (Wikipedia)	1. How many users did Discord reach in July of 2016? 2. What is the monthly subscription fee?	1. 11 million 2. $4.99
Reddit (Wikipedia)	1. When did Reddit claim to have acquired Team Fortress 2? 2. Where is Reddit based?	1. April Fools' Day 2013 2. San Francisco, California

Spotify (Wikipedia)	1. Canvas is only available for what apps? 2. Who believed Spotify users on the app store were Apple's customers?	1. iOS and Android 2. Apple
Strava (Wikipedia)	1. What is the Suffer Score used for? 2. When did Strava switch to Mapbox maps and imagery?	1. Training Plans 2. July 2015

Appendix B

Results of Zero-Shot Experiments of LMs on CoQA and QuAC after Training on SQuAD:

	Evaluation (F1, EM)					
	SQuAD		CoQA		QuAC	
LM	F1	EM	F1	EM	F1	EM
BERT	88.237	80.851	40.264	28.623	26.857	10.552
DistilBERT	85.339	77.010	37.342	26.732	23.290	8.988
RoBERTa	92.101	85.799	43.576	31.955	31.136	14.142
DistilRoBERTa	88.300	81.296	40.181	29.237	26.496	11.694
ALBERT	90.236	82.706	42.403	30.177	31.305	14.142
MobileBERT	88.937	81.41	41.441	29.663	27.148	10.892

Results of Fine-Tuning Experiments of LMs on CoQA and QuAC:

	Further Fine-tuning on CoQA				Further Fine-tuning on QuAC			
	CoQA		SQuAD		QuAC		SQuAD	
LM	F1	EM	F1	EM	F1	EM	F1	EM
BERT	31.608	15.771	65.851	42.564	41.579	33.805	67.946	45.7
DistilBERT	28.999	14.268	62.899	45.279	41.383	32.132	61.944	39.5
RoBERTa	33.537	15.420	65.851	42.564	52.18	39.829	73.982	52.2
DistilRoBERTa	31.929	16.022	67.214	46.055	46.788	34.770	70.220	48.8
ALBERT	33.609	15.984	63.902	39.054	50.39	39.638	64.489	37.1
MobileBERT	30.73	15.558	65.459	45.449	38.904	30.854	65.051	40.4

Results of Zero-Shot Experiments of LMs on Generated Datasets (Websites):

	ByteDance		Discord		Patsnap	
LM	F1	EM	F1	EM	F1	EM
BERT	81.667	79.167	78.464	57.143	86.933	66.667
DistilBERT	90	87.5	84.579	76.19	85.152	61.905
RoBERTa	85.833	83.333	79.153	61.905	86.133	66.667
DistilRoBERTa	96.548	91.667	90.949	71.429	91.464	76.19
ALBERT	94.167	91.667	79.552	52.381	88.015	71.429
MobileBERT	92.778	87.5	84.314	61.905	90.171	80.952

	Reddit		Ripple		CS4248	
LM	F1	EM	F1	EM	F1	EM
BERT	80.802	68.421	81.946	68.182	92.312	84.685
DistilBERT	86.065	73.684	82.249	68.182	90.266	81.081
RoBERTa	90.1	78.947	87.943	68.182	88.462	81.081
DistilRoBERTa	82.699	68.421	86.224	68.182	93.405	86.486
ALBERT	90.51	78.947	86.515	63.636	90.34	83.784
MobileBERT	91.738	84.211	88.079	63.636	93.548	85.586

Results of Zero-Shot Experiments of LMs on Generated Datasets (Wikipedia Pages):

	ByteDance Wiki		Discord Wiki		Reddit Wiki	
LM	F1	EM	F1	EM	F1	EM
BERT	89.074	80.556	84.473	73.016	83.556	73.099
DistilBERT	87.685	77.778	87.052	73.016	80.879	70.175
RoBERTa	87.422	75	85.3	71.429	83.275	73.684
DistilRoBERTa	91.296	80.556	89.054	77.778	82.476	72.515
ALBERT	89.034	80.556	83.909	69.841	84.775	73.684
MobileBERT	89.907	77.778	84.18	73.016	85.357	71.93

	Spotify Wiki		Strava Wiki	
LM	F1	EM	F1	EM
BERT	88.107	79.577	88.542	75
DistilBERT	88.764	80.986	88.571	75
RoBERTa	90.741	84.507	91.875	87.5
DistilRoBERTa	87.689	80.986	89.792	81.25
ALBERT	91.289	85.211	96.094	87.5
MobileBERT	91.464	85.915	95.238	87.5

Results of Fine-Tuning Experiments of LMs on Generated Datasets (Websites):

	ByteDance		Discord		Patsnap	
LM	F1	EM	F1	EM	F1	EM
BERT	90	87.5	79.552	57.143	86.933	66.667
DistilBERT	90	87.5	82.991	71.429	85.152	61.905
RoBERTa	85.833	83.333	79.153	61.905	86.133	66.667
DistilRoBERTa	96.548	91.667	89.656	66.667	91.464	76.19
ALBERT	94.167	91.667	80.418	52.381	88.015	71.429
MobileBERT	96.944	91.667	87.489	66.667	90.171	80.952

	Reddit		Ripple		CS4248	
LM	F1	EM	F1	EM	F1	EM
BERT	84.721	73.684	81.946	68.182	92.312	84.685
DistilBERT	89.297	73.684	83.159	72.727	90.567	81.982
RoBERTa	90.1	78.947	87.943	68.182	88.462	81.081
DistilRoBERTa	82.699	68.421	86.224	68.182	93.945	87.387
ALBERT	90.51	78.947	86.515	63.636	90.85	82.882

MobileBERT	91.738	84.211	88.079	63.636	93.548	85.586

Results of Fine-Tuning Experiments of LMs on Generated Datasets (Wikipedia Pages):

	ByteDance Wiki		Discord Wiki		Reddit Wiki	
LM	F1	EM	F1	EM	F1	EM
BERT	89.074	80.556	84.473	73.016	83.963	73.684
DistilBERT	87.685	77.778	87.647	74.603	81.297	70.76
RoBERTa	87.422	75	85.3	71.429	82.641	73.099
DistilRoBERTa	91.852	83.333	89.054	77.778	83.227	73.099
ALBERT	86.257	75	83.909	69.841	86.072	74.269
MobileBERT	89.907	77.778	83.298	71.429	85.997	73.684

	Spotify Wiki		Strava Wiki	
LM	F1	EM	F1	EM
BERT	88.107	79.577	88.542	75
DistilBERT	88.059	88.282	88.571	75
RoBERTa	91.361	85.915	91.875	87.5
DistilRoBERTa	87.689	80.986	89.732	81.25
ALBERT	90.672	84.507	96.094	87.5
MobileBERT	91.464	85.915	95.238	87.5

Catastrophic Forgetting: Breakdown of SQuAD Questions by Category for Zero-Shot QA

	SQuAD (Before fine-tuning on CoQA and QuAC)							
	F1							
LM	Who	What	When	Where	Why	Which	How	Others
BERT	91.718	87.847	94.413	85.576	75.641	86.795	89.323	87.131
DistilBERT	88.208	84.638	92.634	81.915	75.905	85.345	87.177	84.075
RoBERTa	94.759	91.745	96.313	89.492	84.552	92.084	92.512	91.501
DistilRoBERTa	91.097	87.768	94.076	86.761	78.528	88.732	89.153	87.932
ALBERT	93.369	89.774	95.925	87.622	81.487	89.546	91.532	90.439
MobileBERT	91.766	88.384	96.037	84.663	79.243	88.745	90.489	87.85

	SQuAD (Before fine-tuning on CoQA and QuAC)							
	EM							
LM	Who	What	When	Where	Why	Which	How	Others
BERT	88.418	80.136	90.948	75.058	49.669	80.879	81.913	78.997
DistilBERT	84.463	75.683	89.511	70.67	49.007	78.022	78.274	73.668
RoBERTa	91.62	85.218	93.822	80.139	64.238	87.473	85.031	83.699
DistilRoBERTa	88.23	80.417	90.661	75.982	59.603	83.516	81.289	80.094
ALBERT	90.113	82.304	92.96	78.984	58.94	83.956	83.888	81.818
MobileBERT	87.571	80.483	92.96	74.596	56.954	83.077	83.056	78.683

Catastrophic Forgetting: Breakdown of SQuAD questions by Category After Fine-Tuning on CoQA

	SQuAD (After fine-tuning on CoQA)							
	F1							
LM	Who	What	When	Where	Why	Which	How	Others
BERT	73.96	68.215	69.383	54.119	66.506	76.861	64.056	59.352
DistilBERT	73.506	62.662	68.186	50.74	51.125	71.758	62.404	52.606
RoBERTa	75.005	66.515	62.989	50.774	66.785	76.099	63.612	56.029
DistilRoBERTa	75.907	67.733	68.148	52.569	62.79	77.354	63.289	57.941
ALBERT	73.383	65.024	63.092	54.159	64.824	75.661	62.15	54.019
MobileBERT	74.854	66.189	68.646	47.673	55.322	74.865	60.344	56.112

	SQuAD (After fine-tuning on CoQA)							
	EM							
LM	Who	What	When	Where	Why	Which	How	Others
BERT	59.134	48.999	49.569	24.249	32.45	64.396	43.451	41.379
DistilBERT	61.488	44.264	51.868	26.559	20.53	59.341	44.699	36.52
RoBERTa	57.062	42.824	37.213	19.169	24.503	60.22	41.892	35.58
DistilRoBERTa	62.053	48.287	45.546	23.326	23.841	63.956	41.788	40.125
ALBERT	55.085	40.821	35.92	21.247	26.49	60.879	39.709	33.856
MobileBERT	58.098	46.631	46.408	19.861	17.881	62.418	38.565	36.207

Catastrophic Forgetting: Breakdown of SQuAD questions by Category After Fine-Tuning on QuAC

	SQuAD (After fine-tuning on QuAC)							
	F1							
LM	Who	What	When	Where	Why	Which	How	Others
BERT	71.767	68.148	70.124	61.24	65.884	73.785	61.774	71.47
DistilBERT	70.748	62.032	68.052	56.323	60.675	67.848	49.442	62.929
RoBERTa	80.645	74.688	76.356	66.089	71.699	82.74	62.629	74.113
DistilRoBERTa	76.177	68.459	73.575	59.248	69.251	79.221	63.227	69.539
ALBERT	73.821	64.887	67.139	58.306	65.755	76.462	49.176	64.634
MobileBERT	70.712	65.547	67.367	56.324	66.812	74.112	53.298	66.536

	SQuAD (After fine-tuning on QuAC)							
	EM							
LM	Who	What	When	Where	Why	Which	How	Others

BERT	54.049	46.085	47.701	31.871	29.801	56.923	35.655	53.292
DistilBERT	55.461	39.613	47.701	27.714	20.53	51.868	20.894	41.693
RoBERTa	65.16	53.567	54.31	36.721	33.113	70.11	33.368	53.605
DistilRoBERTa	62.618	48.105	52.155	31.178	33.113	66.813	37.63	51.724
ALBERT	52.825	38.04	37.644	23.095	27.152	58.462	15.177	38.245
MobileBERT	51.036	41.5	43.103	24.942	23.841	56.923	22.869	42.79

Breakdown of Generated Datasets' Questions into Categories after Fine-Tuning on the Respective Generated Dataset's Training Dataset:

1. ByteDance

	Bytedance							
	F1	EM	F1	EM	F1	EM	F1	EM
LM	Who (4)		What (15)		Where (3)		How (2)	
BERT	100	100	84	80	100	100	100	100
DistilBERT	100	100	97.333	93.333	33.333	33.333	100	100
RoBERTa	100	100	77.333	73.333	100	100	100	100
DistilRoBERTa	100	100	94.476	86.667	100	100	100	100
ALBERT	100	100	97.333	93.333	66.667	66.667	100	100
MobileBERT	100	100	97.333	93.333	100	100	83.333	50

2. Discord

	Discord					
	F1	EM	F1	EM	F1	EM
LM	Who (1)		What (18)		Where (1)	
BERT	50	0	82.811	66.667	80	0
DistilBERT	50	0	89.601	83.333	80	0
RoBERTa	100	100	81.235	61.111	100	100
DistilRoBERTa	50	0	93.21	77.778	80	0
ALBERT	50	0	83.821	61.111	80	0
MobileBERT	100	100	92.07	72.222	80	0

3. Patsnap

	Patsnap					
	F1	EM	F1	EM	F1	EM
LM	Who (2)		What (17)		How (2)	
BERT	100	100	83.858	58.824	100	100
DistilBERT	100	100	81.658	52.941	100	100
RoBERTa	100	100	82.87	58.824	100	100
DistilRoBERTa	100	100	89.455	70.588	100	100
ALBERT	100	100	85.195	64.706	100	100
MobileBERT	100	100	87.858	76.471	100	100

4. Reddit

	Reddit			
	F1	EM	F1	EM
LM	What (18)		How (1)	
BERT	83.872	100	72.222	100
DistilBERT	88.702	100	72.222	100
RoBERTa	89.55	100	77.778	100
DistilRoBERTa	81.738	100	66.667	100
ALBERT	89.983	100	77.778	100
MobileBERT	91.279	100	83.333	100

5. Ripple

	Ripple					
	F1	EM	F1	EM	F1	EM
LM	Who (1)		What (17)		How (4)	
BERT	100	100	77.813	64.706	95	75
DistilBERT	100	100	78.205	64.706	95	75
RoBERTa	50	0	88.514	70.588	95	75
DistilRoBERTa	50	0	86.29	70.588	95	75
ALBERT	100	100	83.725	58.824	95	75
MobileBERT	100	100	85.749	58.824	95	75

6. ByteDance Wiki

	ByteDance Wiki									
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
LM	Who (10)		What (9)		When (10)		Where (3)		How (4)	
BERT	90	90	82.963	66.667	90	90	100	100	90	50
DistilBERT	90	90	94.074	77.778	90	90	66.667	66.667	77.5	25
RoBERTa	90	90	91.852	66.667	90	90	70.175	66.667	77.5	25
DistilRoBERTa	90	90	94.074	77.778	90	90	100	100	90	50
ALBERT	85	80	88.519	66.667	90	90	76.19	66.667	82.5	50
MobileBERT	**95**	90	91.852	66.667	90	90	66.667	66.667	90	50

7. Discord Wiki

	Discord Wiki									
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
LM	Who (1)		What (45)		When (6)		Why (1)		How (10)	

BERT	100	100	81.818	68.889	100	100	60	0	88	80
DistilBERT	100	100	86.483	71.111	100	100	50	0	88	80
RoBERTa	100	100	83.568	66.667	100	100	100	100	81.333	70
DistilRoBERTa	100	100	87.342	73.333	100	100	100	100	88	80
ALBERT	100	100	80.139	62.222	100	100	100	100	88	80
MobileBERT	100	100	80.074	64.444	100	100	100	100	84.444	80

8. Reddit Wiki

	Reddit Wiki							
	F1	EM	F1	EM	F1	EM	F1	EM
LM	Who (17)		What (90)		When (32)		Where (6)	
BERT	93.529	88.235	84.824	72.222	86.369	81.25	78.148	50
DistilBERT	96.078	94.118	79.56	68.889	88.125	81.25	78.148	50
RoBERTa	96.471	94.118	84.947	74.444	83.281	78.125	78.148	50
DistilRoBERTa	91.176	88.235	84.271	74.444	86.369	81.25	61.481	33.333
ALBERT	93.529	88.235	87.358	73.333	89.494	84.375	78.148	50
MobileBERT	85.294	82.353	85.631	70	91.25	84.375	84.656	66.667

	Reddit Wiki							
	F1	EM	F1	EM	F1	EM	F1	EM
LM	Why (1)		Which(2)		How (21)		Others(1)	
BERT	100	100	100	100	72.857	66.667	8.333	0
DistilBERT	66.667	0	100	100	69.683	57.143	26.667	0
RoBERTa	0	0	100	100	71.429	57.143	0	0
DistilRoBERTa	57.143	0	100	100	72.222	57.143	100	100
ALBERT	100	100	100	100	74.444	61.905	7.407	0
MobileBERT	61.538	0	100	100	80.794	71.429	100	100

9. Spotify Wiki

	Spotify Wiki							
	F1	EM	F1	EM	F1	EM	F1	EM
LM	Who (4)		WhatDF(77)		When (26)		Where (4)	
BERT	100	100	85.204	75.325	96.795	92.308	89.286	75
DistilBERT	100	100	85.931	77.922	96.795	92.308	75	75
RoBERTa	100	100	87.099	79.221	96.795	92.308	100	100
DistilRoBERTa	100	100	84.092	76.623	96.795	92.308	75	75
ALBERT	100	100	86.159	77.922	96.795	92.308	100	100
MobileBERT	100	100	89.107	81.818	94.103	88.462	75	75

	Spotify Wiki					
	F1	EM	F1	EM	F1	EM
LM	Which(2)		How (28)		Others (1)	
BERT	100	100	84.881	75	100	100
DistilBERT	100	100	86.667	75	44.444	0
RoBERTa	50	50	98.214	96.429	100	100
DistilRoBERTa	100	100	87.857	78.571	100	100
ALBERT	50	50	99.286	96.429	44.444	0
MobileBERT	50	50	99.286	96.429	100	100

10. Strava Wiki

	Strava Wiki							
	F1	EM	F1	EM	F1	EM	F1	EM
LM	Who (1)		What (9)		When (4)		How (2)	
BERT	100	100	79.63	55.556	100	100	100	100
DistilBERT	100	100	79.683	55.556	100	100	100	100
RoBERTa	100	100	85.556	77.778	100	100	100	100
DistilRoBERTa	100	100	85.45	77.778	100	100	100	100
ALBERT	100	100	93.056	77.778	100	100	100	100
MobileBERT	100	100	91.534	77.778	100	100	100	100

11. CS4248 website

	CS4248							
	F1	EM	F1	EM	F1	EM	F1	EM
LM	Who (2)		What (80)		Where (3)		How (26)	
BERT	100	100	90.688	81.25	100	100	95.833	92.308
DistilBERT	80	50	88.765	78.75	100	100	95.833	92.308
RoBERTa	100	100	85.658	76.25	100	100	94.872	92.308
DistilRoBERTa	100	100	92.849	85	100	100	96.154	92.308
ALBERT	100	100	88.693	78.75	100	100	95.726	92.308
MobileBERT	100	100	92.298	82.5	100	100	96.154	92.308