Dialectal Extractive Question Answering (DialQA)

Adel Alkhamisy
Department of Computer Science
George Mason University
aalkhami@gmu.edu

Aniket Pandey
Department of Computer Science
George Mason University
apandey7@gmu.edu

Pankaj Jatav
Department of Computer Science
George Mason University
pjatav@gmu.edu

Yash Bagal
Department of Computer Science
George Mason University
ybagal@gmu.edu

1 Introduction

1.1 Task / Problem

A natural language Question Answering system takes questions in natural language and outputs answer. In this system, there is interaction between the system and the user. This assists in establishing a connection for finding accurate results more quickly. The extractive approach-based QA system comprises of a reader and retriever. Instead of storing the answers to the questions in database, the system loads the document related to the user's question and then the reader extracts the answer. The current QA systems do not consider the faults that might be introduced by speech recognition models and do not take the dialect in to consideration. Through this work, our aim is to make a QA system which understands various languages and then provide a result. In order to achieve this we implement a extractive QA system. The Dialectal Extractive Question Answering systems is built on the existing QA benchmarks TyDi QA (Clark et al., 2020) and SD QA (Faisal et al., 2021). We make use of parts of the SD QA dataset, with recorded dialectal variations of TyDi QA.

1.2 Motivation

In recent times QA systems are a rapidly growing field of study and the main motivation behind them is to answer the human question prompts which not only support the users to find answers but also aid users with visual and motor impairments to interact with devices.

The majority of evaluation benchmarks of existing QA systems rely on text-based which are noise-free. However, the real word usage of such systems involves input with noise. Faisal et al. proposes a QA system based on the SD-QA dataset that aims to mitigate the effects of multi-dialect of users on the QA system. SD-QA covers five languages and twenty-four dialects. However, the user accent is different even among users who speak the same dialect and this challenge is one of the limitations of this study. Another gap in this study is the difference between reading speech and spontaneous speech (Faisal et al., 2021). Also, (Batliner et al., 1995) has recorded readings of text questions that have different characteristics of spontaneous speech. In order to enable researchers to create multilingual models that are effective across several languages Clark et al. proposed large-scale multilingual corpora. However, as the content is not written in various languages, cross-language answer retrieval and translation are required.

1.3 Proposed Approach

We implements a minimal BERT model with attention mechanism for question answering task. The dataset used is SD-QA (Faisal et al., 2021) which is built on top of TyDi QA (Clark et al., 2020).

Question: How many parameters does BERT-large have?
Context: BERT large is really big… it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.24GB, so expect it to take couple of minutes to load on your instance.
Answer: 340M

Question: من قام بلعب دور هاري بوتر في الأفلام المصورة؟
Context: دانيال رادكليف في دور هاري بوتر ، وهو يتيم يبلغ من العمر 12 عامًا ، يقيم عند عمته وعمه الذي لا يرحم ، مشهور بسبب أنه نجا من محاولة قتل على يد الساحر المظلم لورد فولدمورت عندما كان رضيعا، لكن نجح فولدمورت في قتل أبويه ،و هو طالب في مدرسة هوجورتس للسحر[1][1]
Answer: دانيال رادكليف

Figure 1: Illustration of the scenario for QA system that our model aims to evaluate (example from English and Arabic).

Specifically, there are two task which the paper pose Passage Selection Task and Minimal Answer Task. We reduced the maximum sequence length for memory GPU memory optimization. Using our model we perform Extractive-QA using dialectal questions (Speech to text Outputs). We use the Google Speech API to perform speech to text conversion.

2 Approach

2.1 BERT

BERT (Devlin et al., 2019) stands for Bidirectional Encoder Representations from Transformers developed by researchers at Google in 2018, it is based on Transformers, a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection.

It is designed to pre-train deep bidirectional representations from an unlabeled text by jointly conditioning on both the left and right contexts. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of NLP tasks.

2.1.1 Why BERT?

BERT can better understand long term queries and as a result surface more appropriate result. BERT models are applied to both organic search results and featured snippets. While you can optimize for those queries, you cannot "optimize for BERT."

2.2 Model Architecture and Design

Our approach to implement the baseline model is simple, we pass two parameters to the BERT, the input question, and passage as a single packed sequence. The input embeddings are the sum of the token embeddings and the segment embeddings.

1. Token embeddings: A [CLS] token is added to the input word tokens at the beginning of the question and a [SEP] token is inserted at the end of both the question and the paragraph.

2. Segment embeddings: A marker indicating Sentence A or Sentence B is added to each token. This allows the model to distinguish between sentences. In the below example, all tokens marked as A belong to the question, and those marked as B belong to the paragraph.

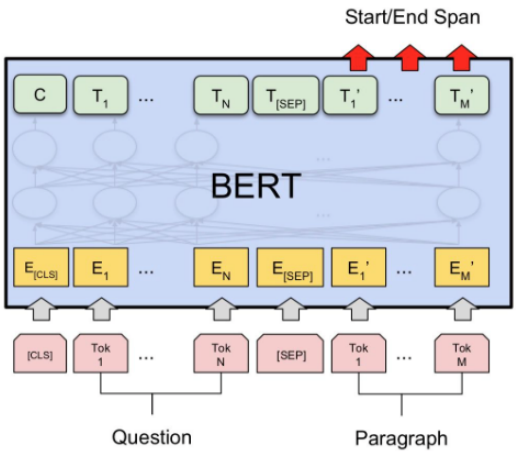

Figure 2: Architecture

For every token in the text, we feed its final embedding into the start token classifier. The start token classifier only has a single set of weights which applies to every word.

After taking the dot product between the output embeddings and the 'start' weights, we apply the softmax activation to produce a probability distribution over all the words. Whichever word has the highest probability of being the start token is the one that we pick.

2.3 Design Setup

To build the complete working architecture we first obtained the gold dataset from SD-QA. In our next step, we used the BERT method of pre-training language representations which obtains state-of-the-art results on a wide array of Natural Language Processing tasks. The BERT open source model was obtained from the Google Research GitHub, BERT repository, see https://github.com/google-research/bert.

Once our complete setup was done, we used the gold passage baseline by https://github.com/google-research-datasets/tydiqa/tree/master/gold_passage_baseline with some minor modification to run the BERT model. We used gold passage task because it is simplified and only the gold answer passage is provided.

3 Experiments

3.1 Datasets

We obtained the Question Answering Dataset which is available on GitHub. See https://github.com/ffaisal93/SD-QA for instructions. The SD-QA (Faisal et al., 2021) dataset extends the TyDi QA (Clark et al., 2020) dataset by incorporating questions, contexts, and responses from five typologically distinct languages. The SD QA extends TyDi QA by adding two dimensions to TyDi-QA. The first dimension is the spoken utterances (questions) to mimic real-world scenarios when the user asks a question to assistant devices like Apple's Siri, and the second dimension is the language dialect.

It is a spoken multilingual and multi-dialect Question Answering dataset. SD-QA contains 24 dialects spread among five languages (Arabic, Bengali, English, Kiswahili, Korean) with more than 68,000 audio prompts generated by 255 annotators. Specifically, SD-QA has development and test data for 11 varieties of English, 7 varieties of Arabic, 2 varieties of Kiswahili, 2 varieties of Bengali, and 2 varieties of Korean.

The dataset has following languages and varieties.

Language	Locations (Variety Code)
Arabic	Algeria (DZA), Bahrain (BHR), Egypt (EGY), Jordan (JOR), Morocco (MAR), Saudi Arabia (SAU), Tunisia (TUN)
Bengali	Bangladesh-Dhaka (BGD), India-Kolkata (IND)
English	Australia (AUS), India-South (IND-S), India-North (IND-N), Ireland (IRL), Kenya (KEN), New Zealand (NZL), Nigeria (NGA), Philippines (PHI), Scotland (SCO), South Africa (ZAF), US-Southeast (USA-SE)
Korean	South Korea-Seoul (KOR-C), South Korea-South (KOR-SE)
Kiswahili	Kenya (KEN), Tanzania (TZA)

Table 1: Languages and sample collection locations in SD-QA dataset.

3.2 Baseline

As a baseline we compared our model to existing QA benchmarks TyDi-QA and SD-QA (Faisal et al., 2021). In our experiment, we used as SD-QA: Spoken Dialectal Question Answering for the Real World (Faisal et al., 2021) as a dataset which is an Extractive-QA model. It uses a multilingual BERT (Devlin et al., 2019) uncased based on huggingface. Multilingual BERT uncased is a single pretrained model capable of handling multiple languages, and pretrained on 102 languages on Wikipedia. The task posed by the SD-QA benchmark is simple; Given a question, and a passage of text containing the answer, mBERT needs to highlight the span of text corresponding to the correct answer. Additionally, SD-QA utilizes Automatic Speech Recognition (ASR) to convert spoken questions to texts using use the Google Speech API.

3.3 Metrics

We use the F1 score to evaluate our model against the exisiting baseline QA benchmark.

3.4 Baseline Results and Discussion

In (Faisal et al., 2021) pre-trained BERT model bert-base-multilingual-uncased is used. During our experiments we found that Multilingual (uncased) has normalization issues in many languages. We used the pre-trained BERT model bert-base-multilingual-cased in contrast to bert-base-multilingual-uncased. On using Multilingual (cased) we were able to resolve the normalization issue and it is recommended in languages with non-Latin alphabets (and is often better for most languages with Latin alphabets). When using this model, make sure to pass `--do_lower_case=false` to run scripts. See Table 2 for the F1 score comparison.

3.5 Fine-Tuning

As our next step for improvement we fine-tuned the learning rate, epoch, and batch size to improve the performance and following were the observations:

3.5.1 Learning Rate

We executed our training of multiple parameters of learning rate [3e-1, 3e-3, 3e-5, 3e-7] and found that the model performs best on *lr=3e-5* with an average F1 score of 72.72 on all languages.

We also observed that the result were drastically low when the learning rate was high and performed best only when *lr=3e-5* which can be observed in Figure 3.

Language	Baseline	Our Result
English-Nigeria	73.36	73.33
English-US	74.35	74.08
English-South India	72.22	71.06
English-Australia	73.67	73.86
English-Philippines	73.76	73.56
Average English	73.47	73.17
Arabic-Algeria	71.72	72.91
Arabic-Egypt	72.39	72.90
Arabic-Jordan	73.27	**75.01**
Arabic-Tunisia	73.55	**73.82**
Average Arabic	72.73	**73.66**
Kiswahili-Kenya	72.12	72.12
Kiswahili-Tanzania	70.74	**71.04**
Average Kiswahili	71.43	**71.57**
All Language	72.60	**72.72**

Table 2: Comparison of F1 score Baseline vs Ours

3.5.2 Epoch

In another approach we experimented with the epoch size and observed that the results almost remain same and fluctuates between 71.02%-72.72%

3.5.3 Batch Size

We also experimented with the batch size ranging from 2 to 16 in order [2, 4, 6, 8, 16] and the overall average result fluctuated between 69.92%-72.72%. However, on performing multiple experiments and fine-tuning the parameters we found substantially good result on three variations of languages namely Arabic Jordan `(arabic-jor)`, Arabic Tunisia `(arabic-tun)`, and Kiswahili Tanzania `(kiswahili--tanzania)` with an increase of **1.74%**, **0.27%**, and **0.27%**

On keeping the batch size 16 the changes observed are.

Language	Baseline	Our Result
Arabic Jordan	73.27	**75.01**
Arabic-Tunisia	73.55	**73.82**
Kiswahili Tanzania	70.74	**71.01**

Table 3: F1 comparison between Baseline and Multilingual Cased model with Batch Size 16.

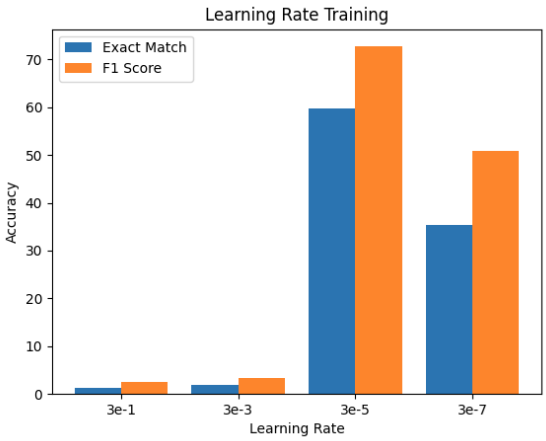

Figure 3: Learning Rate vs Accuracy

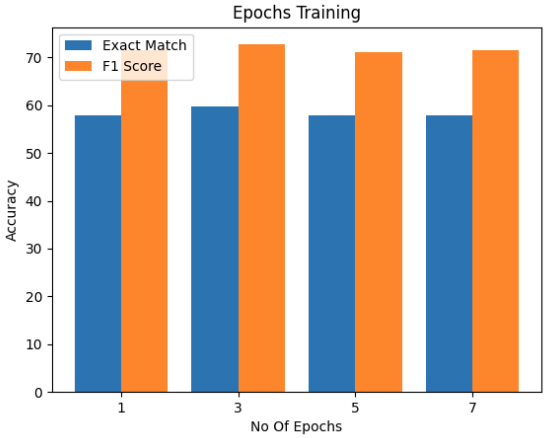

Figure 4: Epoch vs Accuracy

3.6 Results with Pre-Trained Model

We also experimented with various pre-trained models which be obtained from hugging face. Alongside using the pretrained model we also used our custom AdamW optimizer.

1. bert_uncased_L-12_H-768_A-12: The model is a set of 24 BERT models referenced in Well-Read Students Learn Better: On the Importance of Pre- training Compact Models. They are most effective in the context of knowledge distillation, where the fine-tuning labels are produced by a larger and more accurate teacher. This model gave us F1 63.03.

2. Roberta: We used roberta-base-squad2 fine-tuned using the SQuAD2.0 (Lee et al., 2018b) dataset. It's been trained on question-answer pairs, including unanswerable questions, for the task of Question Answering.

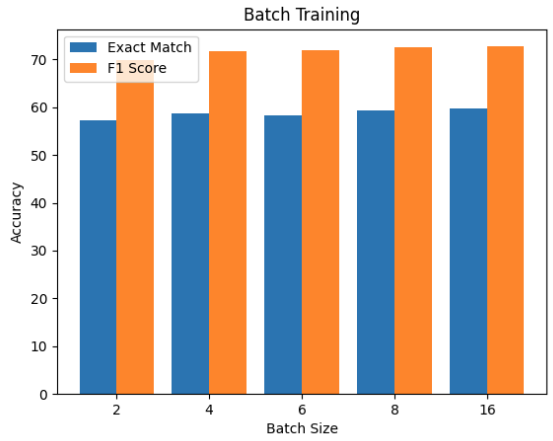

Figure 5: Batch Size vs Accuracy

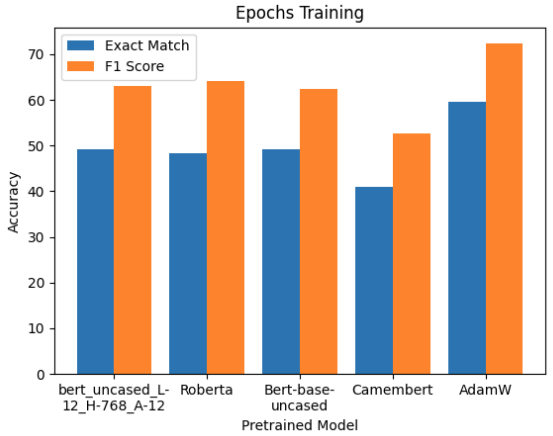

Figure 6: Pre Trained Models vs Accuracy

We obtained F1 64.02 which is better than the previous pre-trained model.

3. bert-base-uncased: It is pretrained model on English language using a masked language modeling (MLM) objective. This model is uncased: it does not make a difference between english and English and gave us F1 62.49.

4. Camembert: We used camembert-base-squadFR-fquad-piaf question-answering French model fine-tuned on a combo of three French Q&A datasets giving us 52.65 F1 result. Our dataset doesn't contain French language however we wanted to experiment by using a pretrained model on a different language and as expected the score dropped significantly.

5. AdamW: The hyper-parameters β_1 and β_2 of AdamW are initial decay rates used when estimating the first and second moments of the gradient, which are multiplied by themselves (exponentially) at the end of each training step (batch). The epsilon is to avoid divide by zero error in the above equation while updating the variable when the gradient is almost zero. So, ideally epsilon should be a small value. But, having a small epsilon in the denominator will make larger weight updates and with subsequent normalization larger weights will always be normalized to one. So we try to alter these three values in our code with β = [0.9, 0.98], ϵ = 1e-9 and obtained 72.29 F1.

Model	Baseline
bert_uncased_L-12_H-768_A-12	63.03
Roberta	64.02
bert-base-uncased	62.49
Camembert	52.65
AdamW	72.29

Table 4: F1 comparison between pre-trained models.

3.7 Visualization of Attention

We also visualized the attention of our BERT uncased model which gave us the highest accuracy.

3.7.1 Head View

The attention in one or more heads from a single Transformer layer. Each line shows the attention from one token (left) to another (right). Line weight reflects the attention value (ranges from 0 to 1), while line color identifies the attention head. See Figure 7 for illustration.

3.7.2 Model View

The model view provides a birds-eye view of attention throughout the entire model. Each cell shows the attention weights for a particular head, indexed by layer (row) and head (column). The lines in each cell represent the attention from one token (left) to another (right), with line weight proportional to the attention value (ranges from 0 to 1). See Figure 8 for detailed view.

3.7.3 Neuron View

The neuron view visualizes the intermediate representations to compute attention. The traces of the chain of computations that produce these attention weights are shown in Figure 9.

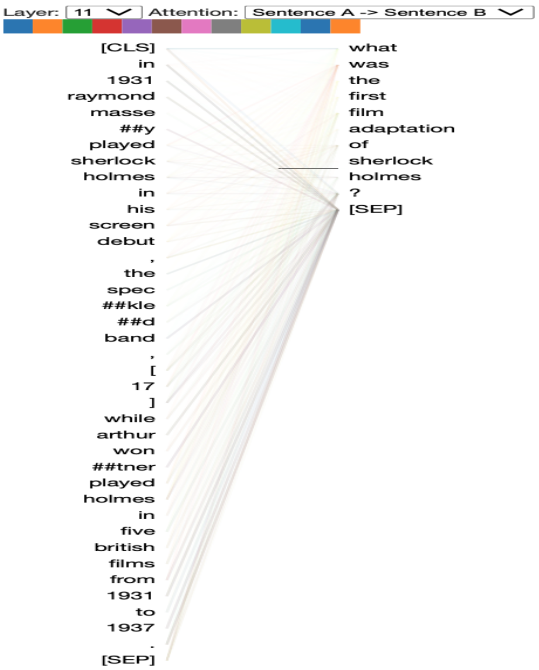

Figure 7: Head View

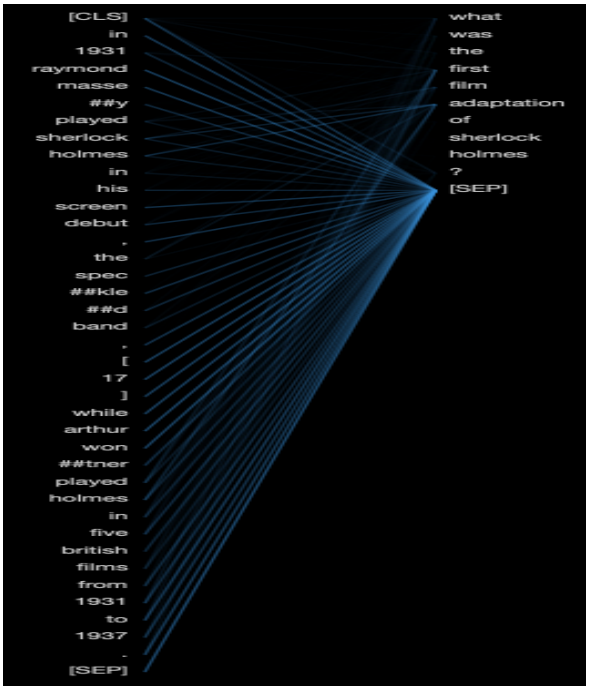

Figure 9: Neuron View

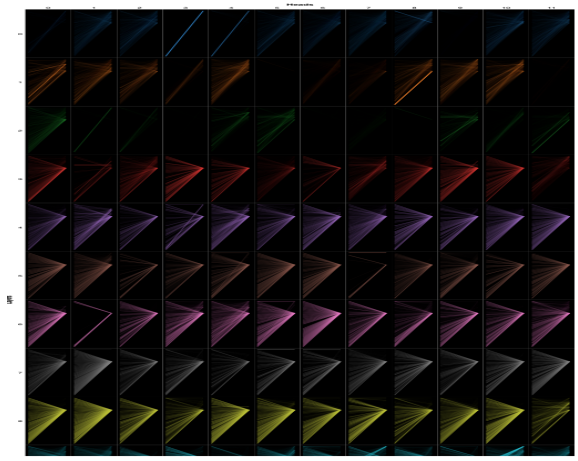

Figure 8: Model View

4 Related work

In recent times QA systems are a rapidly growing field of study and the main motivation behind them is to answer the human question prompts which not only support the users to find answers but also aid users with visual and motor impairments to interact with devices.

In recent times many related work have been done to take us from text based QA to voice based QA machines. In the work by Ravichander et al. in 2021 they talk about the introduction of noise and error in QA system by keyboard input, machine translation and speech input. The paper discusses about tests performed which shows that missing punctuation degrade the result by 5.1%. Other factor such as accent is also taken into consideration. There is another similar work in (Peskov et al., 2019) which studies mitigating ASR errors in QA, assuming white-box access to the ASR systems.

The latest work is done by Faisal et al.. in which they discuss about errors introduced by speech recognition models and language variations (dialects) of the users. They conducted implementation for improving the existing TyDi QA (Clark et al., 2020) dataset for the formation of a multi-dialect, spoken QA on five different languages. TyDi QA is a question answering dataset covering 11 typologically diverse languages with 204K question answer pairs. About the typology, it is diverse and contains language phenomena that would not be found in English-only corpus.

Further, in another research Spoken SQuAD (Lee et al., 2018b), the QA is done in text form and comprehensive reading in speech form. The speech data is first converted to text and the output could be either text or audio. This was further modified and augmented in ODSQA (Lee et al., 2018a) where the QA was also given in audio form. In these experiments the ASR errors significantly degraded the performance of reading comprehensive models and then they proposed the use of different kinds of sub-word units to mitigate the impact of ASR errors. The other work also tried

to bridge the gap between Question Answer and dialog by answering various questions using conversational manner.

5 Conclusion

We propose a dialectal extractive robust question answering system which uses the pretrained model BERT to achieve baseline on SD-QA with the limited GPU memory space by reducing the max-seq-length. We tested multiple pre-trained models and bert-base-multilingual-cased outperforms all other models on the same setting.

We hope that this baseline can constitute a good starting point for researchers wanting to create better and robust multilingual models for the Natural Questions Answering and for other question answering dataset with similar characteristics.

While working on this task we came across many pre-trained models but due to limited time we could not test all of the models and hyperfine each model for this dataset. Below are the some tasks that we want to try with dataset in future.

1. ByT5 (Xue et al., 2021): Towards a token-free future with pre-trained byte-to-byte models. We want to hyperfine and see how much accuracy we can improve.

2. MetaFormer (Yu et al., 2022): This model have been used for images but we want to use the same Transformer for this dataset and check the results.

3. BERT with Language-Clustered vocabulary (Chung et al., 2020)

References

A. Batliner, R. Kompe, A. Kießling, H. Niemann, E. Nöth, A.J.R. Ayuso, and J.M.L. Soler. 1995. *Can You Tell Apart Spontaneous and Read Speech If You Just Look at Prosody?* NATO ASI Series. Universität Augsburg.

Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving multilingual models with language-clustered vocabularies.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

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.

Fahim Faisal, Sharlina Keshava, Md Mahfuz Ibn Alam, and Antonios Anastasopoulos. 2021. SD-QA: Spoken dialectal question answering for the real world. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3296–3315, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chia-Hsuan Lee, Shang-Ming Wang, Huan-Cheng Chang, and Hung-Yi Lee. 2018a. Odsqa: Open-domain spoken question answering dataset.

Chia-Hsuan Lee, Szu-Lin Wu, Chi-Liang Liu, and Hung yi Lee. 2018b. Spoken SQuAD: A Study of Mitigating the Impact of Speech Recognition Errors on Listening Comprehension. In *Proc. Interspeech 2018*, pages 3459–3463.

Denis Peskov, Joe Barrow, Pedro Rodriguez, Graham Neubig, and Jordan Boyd-Graber. 2019. Mitigating noisy inputs for question answering.

Abhilasha Ravichander, Siddharth Dalmia, Maria Ryskina, Florian Metze, Eduard Hovy, and Alan W Black. 2021. Noiseqa: Challenge set evaluation for user-centric question answering.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2021. Byt5: Towards a token-free future with pre-trained byte-to-byte models.

Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. 2022. Metaformer baselines for vision.