NoiseQA: Challenge Set Evaluation for User-Centric Question Answering

Abhilasha Ravichander **Siddharth Dalmia** **Maria Ryskina**
Florian Metze **Eduard Hovy** **Alan W Black**
Language Technologies Institute, Carnegie Mellon University, USA
{aravicha,sdalmia,mryskina}@cs.cmu.edu

Abstract

When Question-Answering (QA) systems are deployed in the real world, users query them through a variety of interfaces, such as speaking to voice assistants, typing questions into a search engine, or even translating questions to languages supported by the QA system. While there has been significant community attention devoted to identifying correct answers in passages assuming a perfectly formed question, we show that components in the pipeline that precede an answering engine can introduce varied and considerable sources of error, and performance can degrade substantially based on these upstream noise sources even for powerful pre-trained QA models. We conclude that there is substantial room for progress before QA systems can be effectively deployed, highlight the need for QA evaluation to expand to consider real-world use, and hope that our findings will spur greater community interest in the issues that arise when our systems actually need to be of utility to humans.[1]

1 Introduction

Everyday users now benefit from powerful QA technologies in a range of consumer-facing applications including health (Jacquemart and Zweigenbaum, 2003; Luo et al., 2015; Abacha and Demner-Fushman, 2016; Kilicoglu et al., 2018; Guo et al., 2018), privacy (Sathyendra et al., 2017; Harkous et al., 2018; Ravichander et al., 2019), personal finance (Alloatti et al., 2019), search (Yang, 2015; Bajaj et al., 2016; He et al., 2018; Kwiatkowski et al., 2019) and dialog agents (Dahl et al., 1994; Raux et al., 2005). Voice assistants such as Amazon Alexa[2] or Google Home[3] have brought natural language technologies to several million homes globally (Osborne, 2016; Jeffs, 2018). Yet, even with millions of users now interacting with these technologies on a daily basis, there has been surprisingly little research attention devoted to studying the issues that arise when people use QA systems.

Traditional QA evaluations do not reflect the needs of many users who *can* benefit from QA technologies. For example, users with a range of visual and motor impairments now rely extensively on voice interfaces (Pradhan et al., 2018) for efficient text entry.[4] Another need is cross-lingual information access, e.g. in scenarios where a speaker of one of the ∼7000 non-English living languages in the world (Eberhard et al., 2020) may want to take advantage of an English QA system.[5] QA evaluation has to keep up with the different ways in which users may use these systems in practice, and the different users who interact with these systems.

Keeping these needs in mind, we construct evaluations considering the *interfaces* through which users interact with QA systems.[6] We analyze errors introduced by three interface types that could be connected to a QA engine: **speech recognizers** converting spoken queries to text, **keyboards** used to type queries into the system, and **translation systems** processing queries in other languages. Our contributions are as follows:

1. We identify and describe the problem of interface noise for QA systems. We construct a challenge set framework for errors introduced by three kinds of interfaces: speech recognizers, keyboard interfaces, and translation engines, based on the popular SQuAD question-answering benchmark (Rajpurkar et al., 2016). We define synthetic noise generators, as well

[1] All resources available at noiseqa.github.io .
[2] developer.amazon.com/alexa
[3] assistant.google.com
[4] More than 3.4 million American adults over the age of 40 have a form of visual impairment (Congdon et al., 2004).
[5] As of 2021-01-24, there are 6,235,415 articles on English Wikipedia making it the largest edition: wikicount.net
[6] 'QA system' refers to any computing engine that receives a users' question and constructs an answer. It may consist of an end-to-end neural architecture or a structured pipeline.

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 2976–2992
April 19 - 23, 2021.

Original Question	Interface	Synthetic Construction	Natural Construction
What has a Lama determined to do?	ASR	what has a llama determined to do	what has a llama determined to do
What has a Lama determined to do?	Keyboard	Wjat has a Lsma determined yo do?	WHat has a Lama determied to do?
What has a Lama determined to do?	MT	What has a Lama decided to do?	What is a llama determined to do?

Table 1: Example question perturbations from synthetic and natural noise challenge sets for three types of interfaces: Automatic Speech Recognition (ASR) systems, Keyboard and Machine Translation (MT) systems.

as manually construct natural noise challenge sets, by processing SQuAD questions through the specified interfaces.

2. We evaluate the performance of current state-of-the-art methods on natural and synthetic noisy data. We find that accessibility needs to be consciously worked towards, as we see that the performance of QA systems can be impacted by the choice of interface.
3. We analyze the generated noise and its impact on the downstream question answering and conduct an initial exploration of mitigation strategies for interface errors, focusing on data augmentation and query repair.

2 Motivation

Modern QA systems often rely on large databases of digital text such as Wikipedia as their source of knowledge; such corpora typically contain well-formed text in a high-resource language like English. However, the user's input could come in many different forms: it could be spoken, or written but in another, possibly lower-resource language. To convert these inputs into the format that the system can process, another machine learning system such as a speech recognizer or a machine translation engine is required, and these intermediate systems will inevitably propagate their decoding errors into the QA engine. However, interface errors are not necessarily artifacts of machine learning models: even when the question comes in the desired form (e.g. English text), it has to be communicated to the QA system through a mechanical interface such as a keyboard, and the process of typing can introduce errors such as character substitutions. To be useful in real-world settings, a QA system has to be able to correctly process the input question regardless of the input interface. We simulate the use cases for three interface categories (ASR, MT, and keyboard) with different level of human involvement, from fully automatic pipelines to leveraging existing human-generated resources to manual annotation, and evaluate whether the modern QA systems are capable of going from controlled well-formed inputs to real-world scenarios.

3 Challenge Set Construction

We define a suite of three types of noise perturbations, each imitating noise specific to a category of interfaces, and apply them to the data to create the challenge sets. We choose to add the noise to the questions but not to the context paragraphs, to replicate a realistic scenario of the noise being introduced to the question by the interface through which the user interacts with the QA engine. For each type of noise, we both build a synthetic generator that can introduce noise on a large scale, as well as manually create 'natural' noise challenge sets to imitate real-world noise.

Our challenge sets are based on SQuAD 1.1 (Rajpurkar et al., 2016),[7] a large-scale machine comprehension dataset based on Wikipedia articles where the answer to each question is a span in a provided context. We choose SQuAD both for its popularity as a benchmark (Gardner et al., 2018; Devlin et al., 2019; Radford et al., 2018; Wolf et al., 2019) and to avoid additional confounds such as unanswerable questions (Rajpurkar et al., 2018).[8] We use the standard ∼90K/10K train/development split and construct the challenge sets from the XQuAD data (Artetxe et al., 2020), a subset of 1,190 SQuAD development set questions accompanied by professional translations into ten languages.[9] Below we discuss each challenge set in more detail.

[7] Though in principle, these constructions could be applied to any kind of QA dataset

[8] Future work would pursue a context-driven evaluation of unanswerability, identifying the kinds of unanswerable questions users ask in practice (Ravichander et al., 2019; Asai and Choi, 2020).

[9] Spanish, German, Greek, Russian, Turkish, Arabic, Vietnamese, Thai, Chinese, and Hindi.

3.1 MT Noise

Our first challenge set emulates machine translation noise introduced when the question is asked in a language other than the language of the QA system's training data. We use English as the QA system language, pairing English contexts with non-English questions.

Synthetic Challenge Set Our synthetic noise generator employs the back-translation technique (Sennrich et al., 2016; Dong et al., 2017; Yu et al., 2018). In our case, back-translation is not meant to act as a data augmentation technique but rather to simulate noise that could be introduced by an MT engine when translating the question from another language. We imperfectly approximate natural non-English input by automatically translating English questions into a pivot language (German); we then translate them back to English, imitating a scenario where the user submits a query through an MT engine. We use the HuggingFace implementation (Wolf et al., 2019) of MarianNMT (Junczys-Dowmunt et al., 2018).[10]

Natural Challenge Set To bring our simulation closer to the natural setting, we create another challenge set from English machine translations of human-generated questions in other languages. We take the questions from the XQuAD dataset, which consists of English questions paired with professional translations into ten other languages.[11] For each of the test set languages, we use Google's commercial translation engine[12] to produce the English translation of the question. This allows us to construct ten challenge sets of translations from different languages with 1,190 questions each.

3.2 Keyboard Noise

This challenge set represents the noise introduced in the process of typing a question up on a keyboard, for example, when a question is submitted to a QA system through a search engine.

Synthetic Challenge Set Inspired by prior work (Belinkov and Bisk, 2018; Naik et al., 2018), our basic noise generator introduces per-character typos based on the proximity of the keys in a standard QWERTY keyboard layout. Each word is corrupted with a 25% probability by substituting a randomly sampled character with its row-wise neighbor. We also create more natural-looking noise by introducing externally collected human misspellings into our data on word level, as proposed by Belinkov and Bisk (2018). Although prior work refers to this as natural noise, emphasizing that the typos have been produced by humans, we consider it synthetic because the errors are applied to the data outside of their original context. We start with the Wikipedia common English misspellings list[13] and apply a simple filtering heuristic that only retains keyboard errors (see Appendix C), obtaining 1,742 misspellings for 1,489 English words.

ORIGINAL QUESTION	How many Panthers defense players were selected for the Pro Bowl?
GOOGLE ASR	how many Santa's defense players selected for the Pro Bowl
ESPNET (WITH LM)	how many pantols the tent places were slected for the probol
KALDI (WITH LM)	how many friends tons of defence UNK for the UNK

Table 2: Example outputs of different ASR systems on a recorded question from SQuAD (Rajpurkar et al., 2016).

Natural Challenge Set To generate errors specific to the context of the question rather than hypothesized to exist at a lexical level across contexts, we ask three human annotators to retype English XQuAD questions. Annotators can see the original question, which helps avoid errors caused by misconception (e.g. not knowing the correct spelling of a named entity), but not their own input, in order to prevent them from correcting the typos. Of the obtained noisy questions, 51.6% and 25.7% differ from the original by at least one or at least two characters respectively.

3.3 ASR Noise

Our final challenge set simulates ASR errors that occur when a question is posed to a voice interface.

Synthetic Challenge Set We emulate automatic recognition of natural speech by using a Text-to-Speech (TTS) system pipelined with an ASR engine (Tjandra et al., 2017). We voice the questions using Google TTS and transcribe the obtained

[10] huggingface.co/Helsinki-NLP/opus-mt-{en-de|de-en}

[11] A subtle nuance is that XQuAD questions are not originally written in these languages but translated from English; acknowledging this, we use XQuAD data as the natural challenge set because its fully parallel nature allows varying input language while controlling for content for fair comparison.

[12] translate.google.com

[13] en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings

Interface	CER (↓)	WER (↓)	BLEU (↑)
Synthetic			
ASR	3.96	16.61	77.12
Keyboard	4.11	23.93	52.66
Translation	20.51	29.36	58.42
Natural			
ASR	12.96	30.67	57.22
Keyboard	1.78	7.42	85.78
Translation	31.89	43.34	47.07

Table 3: % Character Error Rate (CER), % Word Error Rate (WER) and BLEU scores for all challenge sets compared to ground truth. For ASR and MT, synthetic noise is less prominent than natural, reflecting the idealized simulation conditions. As expected, natural keyboard noise demonstrates the best word-level statistics.

speech using Google Speech-to-Text optimized for English–US. Besides Google ASR, we use Kaldi ASpIRE (Povey et al., 2011; Peddinti et al., 2015) and ESPnet CommonVoice (Watanabe et al., 2018; Ardila et al., 2020) open-source systems, as shown in Table 2. We choose the former for analyzing the downstream effect of out-of-vocabulary word prediction in fixed vocabulary decoding (Peskov et al., 2019) and the latter for data augmentation (§4.2) due to its improved out-of-vocabulary word handling with subword units. In order to generate the large amount of speech data needed for augmentation, we use the open-source ESPnet LJSpeech TTS (Hayashi et al., 2020; Ito and Johnson, 2017) to voice the questions.

Natural Challenge Set We use the SANTLR speech annotation toolkit (Li et al., 2019) to record spoken versions of the prompt question from three human annotators (for background details, see Appendix D). The obtained recordings are then transcribed using the ASR engines listed above. As expected, recognizing human speech is more difficult: the word error rate of the Google ASR system on the obtained set is 31%, compared to 17% on the synthesized English–US speech.

4 Experiments

We select four QA models that demonstrated strong performance on SQuAD 1.1[14] to be tested under interface distortions: BiDAF (Seo et al., 2017), which represents contexts at different levels of granularity using bidirectional attention flow mechanism; its extension BiDAF-ELMo (Peters et al., 2018) augmented with contextualized embeddings; BERT (Devlin et al., 2019), a bidirectional Transformer-based language model (Vaswani et al., 2017); and RoBERTa (Liu et al., 2019), a more robustly pre-trained version of BERT.

4.1 Results and Analysis

Table 3 shows the character error rate (CER), word error rate (WER) and BLEU score[15] for the generated challenge sets. Synthetic ASR and MT pipelines introduce substantially less noise than their natural counterparts, while the opposite holds for the keyboard. This is likely due to the generators not being equally controllable: while we can arbitrarily make the synthetic keyboard set noisier by increasing the corruption rate, synthetic ASR and MT pipelines include black-box components which also make the task easier for the interface by design (TTS synthesizes idealized speech, back-translation mimics MT training conditions).

In this section, we investigate how robust QA models are to these interface errors. Table 4 reports the performance on both synthetic and natural challenge sets. For brevity, we present results using the German–English model and the Google ASR for MT and ASR respectively.

First, we observe that both synthetic and natural noise decrease accuracy for all models and interfaces, with synthetic keyboard and natural ASR errors being the most challenging. As for MT noise, Table 4 reports results on German queries; although the systems seem robust on these, we find that MT noise can actually be quite challenging with sharp degradation of performance on Thai and Arabic (Figure 2). Further, we notice that the relative performance of models on the development set is not necessarily a sufficient proxy for the relative robustness of models to interface errors: while BERT and RoBERTa perform very similarly on XQuAD–English, RoBERTa outperforms BERT on handling all three kinds of interface errors. For practitioners, this could suggest that simply choosing the highest-accuracy QA model without separately evaluating robustness to interface noise may lead to sub-optimal performance in practice.

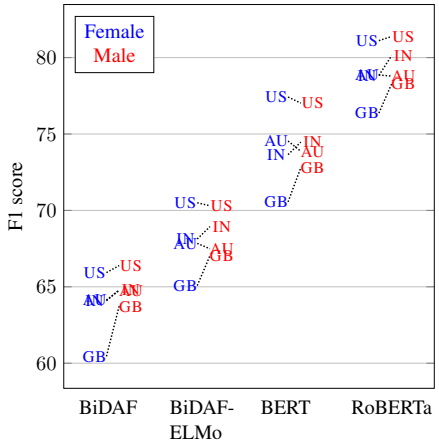
Below we discuss the effect of each interface in more detail.

[14] F1 scores on SQuAD dev set: BiDAF: 77.8; BiDAF-ELMo: 80.7; BERT: 88.8; RoBERTa: 89.9. For hyperparameters and implementation details, see Appendix A.

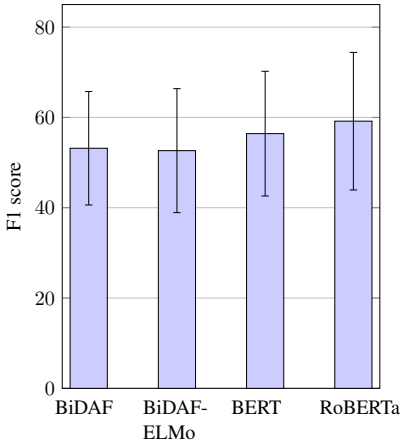
[15] Uncased detokenized BLEU using SacreBLEU (Post, 2018).

Model	$XQuAD_{EN}$ EM	$XQuAD_{EN}$ F1	ASR EM	ASR F1	MT EM	MT F1	Keyboard EM	Keyboard F1
Synthetic								
BiDAF (Seo et al., 2017)	60.08	71.96	54.62	66.39	55.97	68.01	45.21	57.78
BiDAF-ELMo (Peters et al., 2018)	62.61	75.38	56.81	70.30	57.39	70.05	50.93	63.80
BERT (Devlin et al., 2019)	72.77	84.66	61.93	77.02	67.23	79.08	61.76	73.64
RoBERTa (Liu et al., 2019)	72.35	84.42	68.07	81.38	68.40	80.93	65.04	76.97
Natural								
BiDAF (Seo et al., 2017)	60.08	71.96	45.97	57.64	54.87	66.90	56.89	68.33
BiDAF-ELMo (Peters et al., 2018)	62.61	75.38	49.16	62.49	59.24	71.06	60.76	73.32
BERT (Devlin et al., 2019)	72.77	84.66	52.94	67.13	68.82	79.98	69.16	81.84
RoBERTa (Liu et al., 2019)	72.35	84.42	60.08	73.61	70.00	82.13	70.92	83.37

Table 4: Performance of the QA models under the three kinds of interface noise: ASR (using Google ASR), MT (with the German–English model), and keyboard. All models score lower on noisy data, most notably on the natural ASR set. MT noise is less prominent, but we later show its impact is highly dependent on the input language.



(a) F1 scores for TTS voices over 4 accent and 2 gender settings. Lines connect the results for different gender but same accent and model.



(b) F1 mean and standard deviation over 4 human voices. For details and score breakdown by speaker, see Appendix D.

Figure 1: Effect of synthetic (a) and natural (b) voice variation on the QA performance in an ASR pipeline. Synthetic voice variation is achieved by varying accent and gender settings in the Google TTS model; US accent setting shows the highest scores while neither gender setting consistently performs best (indicated by line slopes). Natural variation is measured on a sample of 100 questions narrated by four annotators. All models exhibit considerable variation in both experiments.

ASR Noise: Speech recognizers typically omit punctuation, which could mean losing cues important for the downstream task. To look at this factor in isolation, we remove punctuation from the original XQuAD questions. This change alone decreases BERT performance by 5.1 F1, suggesting that the absence of punctuation in part explains the degradation in the presence of ASR noise. When we qualitatively analyze a sample of 50 questions that BERT answered successfully in the original setting but not when passed through the speech interface, we find that 14% of them are identical to the original modulo punctuation. Other sources of error include the ASR producing completely meaningless questions (28%), hallucinating (12%) or losing named entities (10%), and replacing words with homonyms (4%); other difficult cases include recognizing acronyms and preserving possessives, tense, and number (2% each). Although these problems could be diminished by designing better interfaces, we believe it is also worthwhile for practitioners to work on improving robustness of the QA systems itself: many interfaces, especially commercial, only offer black-box access, and building a completely noise-free interface is not feasible.

Voice variation also plays a role: ASR error distribution differs by speaker background variables such as accent (Zheng et al., 2005), in turn affecting the downstream systems (Harwell, 2018; Lima et al., 2019; Palanica et al., 2019). To emulate speaker variation in the synthetic setting, we use Google English Text-to-Speech to pronounce the XQuAD questions in eight different voices, varying the provided accent and gender settings. As Fig-

ure 1a shows, all models exhibit considerable variation in F1 score, consistently performing best on synthetic US accent (which our speech recognizer is optimized for) and worst on GB. Score breakdowns by setting can be found in Appendix D.

We also repeat the experiment with four human speakers narrating a sample of 100 XQuAD questions, to control for content. As shown in Figure 1b, each model's performance varies substantially between voices. The four speakers differ by accent (2 Indian, 1 Russian, 1 Scottish), gender (2 male, 2 female), and level of proficiency (native and non-native); more details and individual speaker scores can be found in Appendix D.[16] Although improving robustness to accent variation is out of the scope of our work, we highlight that the performance can degrade sharply depending on the user and their acoustic conditions.

We also analyze how the choice of ASR model affects the QA accuracy, focusing in particular on the decoding strategies for out-of-vocabulary words. We compare Kaldi, which outputs an UNK token for unknown words (Peddinti et al., 2015), and Google's large-vocabulary ASR model. On our set of human voices, Kaldi produces at least one UNK token for ∼50% of the questions, and BERT achieves an F1 score of only 43.6 on this set (54.4 F1 and 32.3 F1 separately on questions with and without UNK respectively) compared to 67.1 F1 achieved by Google ASR, demonstrating that speech recognizer choice can greatly affect downstream QA performance. The observed degradation due to UNK decoding (previously noted by Peskov et al., 2019) suggests that practitioners might find it useful to go beyond speech recognition benchmarks, and also evaluate ASR systems in the context of downstream QA applications.

Translation Noise: As Table 4 shows, German–English translation errors affect the performance of all models, although to a lesser extent than ASR noise. However, the MT quality and, in turn, the downstream performance varies greatly depending on the source language. Figure 2 shows BERT and RoBERTa F1 scores on questions translated from each of the ten XQuAD languages to English (numbers reported in Appendix E). While German and Spanish have the highest accuracy, lower-resource and more typologically distant languages like Arabic and Thai are far behind. On translated Thai inputs, BERT achieves only 71.0 F1, which is a 16% drop in accuracy from the original English setting compared to 6% for German.

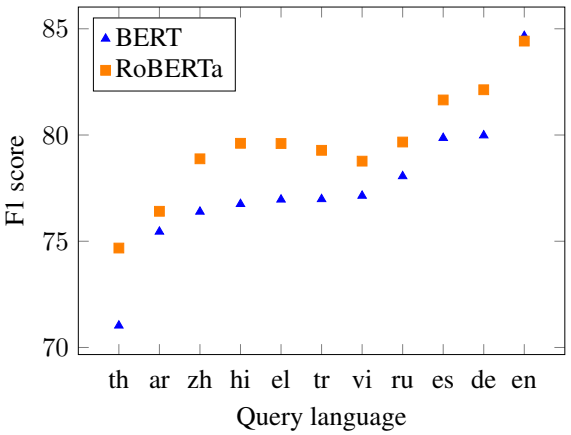


Figure 2: Effect of the input language on the QA system performance in an MT pipeline. Automatically translating non-English queries to English decreases performance for all source languages, and the decrease is especially noticeable for lower-resource languages.

Table 5 shows example translations from four XQuAD languages and highlights their divergences from the original questions. Since the questions are being translated out of context, MT tends to replace important content words with ones that are semantically related but not appropriate in given context (*Lord→deity*, *chair→President*, *ctenophore→jellyfish*). Transliteration of technical terms and named entities is also a challenge, especially for languages written in non-Latin scripts (*ctenophore→tenophora* through Hindi, *Jochi→Dschötschi* through German). For further qualitative analysis, we sample 100 questions translated from Hindi which BERT fails to answer correctly despite accurately answering their English equivalent. Of these, 30% were identified by a native speaker annotator as paraphrases of the original question that would admit the original answer. The remaining incorrect translations are due to question type shift (31%), ungrammatical or meaningless questions (12%), corrupted named entities (8%) and dropped determiners (2%; Hindi does not generally use definite articles). Some divergences also go beyond word level, e.g. 10% of questions have semantic role inversion (*What earlier market did the Grainger Market replace?→Which earlier market replaced Granger's market?*). While some word-level errors can be corrected post-hoc, repairing syntax is much more challenging, which again

[16] Comparisons between demographics should not be drawn from per-speaker results, since we do not control for confounds like recording conditions, aiming for a realistic sample.

Language	Question	Language	Question
What type of Lord is Doctor Who?		**When would the occupation of allies leave Rhineland?**	
de:	What kind of gentleman is Doctor Who?	**de**:	When would the Allied occupation leave the Rhineland?
zh:	What type of lord is Doctor Who?	**zh**:	When was the Allies scheduled to withdraw from Rhineland?
hi:	What kind of deity is Doctor Who?	**hi**:	When will the Rhineland be removed from the occupation of the Allied countries?
ru:	What type of overlord is Doctor Who?	**ru**:	When did the Allies intend to remove the occupation of the Rhine region?
Who is the chair of the IPCC?		**How much food does a ctenophora eat in a day?**	
de:	Who is the chair of the IPCC?	**de**:	How much food does a jellyfish eat in a day??
zh:	Who is the current chairman of the IPCC?	**zh**:	How much food does a jellyfish eat in a day?
hi:	Who is the President of IPCC?	**hi**:	How much food does a tenophora eat in a day?
ru:	Who is the chairman of the IPCC?	**ru**:	How much food does a ctenophore eat per day?

Table 5: Examples of translation divergences for German (de), Chinese (zh), Hindi (hi), and Russian (ru).

brings it down to the robustness of the QA engine.

Keyboard Noise: Synthetic keyboard noise produced by our key-swap typo generator has a much stronger effect on the QA performance than natural noise (11.1 F1 and 2.4 F1 drop respectively). We attribute this to differences in the perturbation intensity: ∼25% of question words are corrupted in the synthetic setting, but only ∼9% of words are corrupted under natural conditions.[17] Interestingly, BiDAF- and BERT-based models consistently show comparable decreases in F1 score, suggesting that character-level tokenization of the former does not on its own guarantee robustness to typos.

Another factor that could affect downstream performance is error placement. We evaluate BERT on three additional synthetic sets, introducing noise to only function words (conjunctions, pronouns, articles), only content words (which we limit to nouns and adjectives), or only commonly misspelled words (using the Wikipedia misspellings list as described in §3.2). Synthetically perturbing all function words and all content words decreases F1 score by 6.7 and 11.7 respectively, confirming that not all words are equally important for the model finding the correct answer. Injecting the interface errors from Wikipedia into the 2,716 questions containing at least one commonly misspelled word yields F1 score of 78.6 (6.1 F1 drop), showcasing the decreased performance we would likely see in real-life user interactions.

4.2 Mitigation Strategies

We experiment with two strategies for improving the QA system robustness: repairing the question errors using the provided context and retraining QA models on the data augmented with synthetic noise. Question repair assumes availability of context, making it unsuitable for open-domain QA, but reasonable for use cases like QA over manuals or policies (Feng et al., 2015; Harkous et al., 2018; Ravichander et al., 2019). This approach treats words that occur in the question but not the context as potential noise, attempting to replace them with the closest candidate from the context paragraph. We use character error rate as the distance metric, empirically setting the threshold to 0.5 using the synthetic set. We perform two experiments, applying the repair either only to content words (here, nouns and adjectives) or only to named entities in both the context and the question. Table 6 shows how these repairs affect BERT performance on three types of natural noise. Named entity repair yields marginal improvements across the board, while content word repair has a stronger effect but only for keyboard errors. The proposed strategy could also be combined with other deterministic or off-the-shelf repair methods, such as adding final question marks for ASR (+6.52 F1) or using a spellchecker for keyboard (+1.41 F1).

For data augmentation, we use our synthetic noise generators to inject noise into ∼90K SQuAD training questions and retrain BERT on the combined clean and noisy data. As Table 6 shows, augmentation yields improvements on all three types of natural noise over BERT trained on clean data only, but the performance of the augmented models drops slightly on the clean data. Best results on natural ASR and MT noise are obtained when the data is augmented with the same type of synthetic noise; interestingly, this is not true for keyboard noise, where ASR augmentation also works best.

[17] Synthetic data corruption rate is a design decision and can be made to simulate the expected natural noise or be more challenging as a stress test, depending on practitioner's goals.

	$XQuAD_{EN}$		ASR		MT		Keyboard	
BERT Model	EM	F1	EM	F1	EM	F1	EM	F1
BERT	72.77	84.66	52.94	67.13	68.82	79.98	69.16	81.84
+ Named entity repair	**72.94**	**84.78**	53.03	67.34	68.82	80.05	69.58	82.22
+ Content word repair	**72.94**	84.77	52.61	67.01	68.32	79.76	70.25	**82.60**
+ Augmentation	72.35	83.89	**64.37**	**75.89**	**68.90**	**80.83**	**70.76**	82.43

Table 6: Effect of question repair and data augmentation on BERT performance on three types of natural noise. Results on synthetic noise and data augmentation score breakdown by interface can be found in Appendix F.

Although our results are preliminary, they suggest that augmentation could prove useful in enabling effective question answering in the real world.

To better understand where ASR and MT augmentation helps, we compare the performance of augmented and baseline BERT on additional challenge sets, synthesizing some common noise artifacts in isolation. We find that ASR noise augmentation improves robustness to omission of punctuation: ASR-augmented model achieves 82.7 F1 on questions with no punctuation and 82.9 F1 on questions without the final question mark (compared to 79.2 and 79.6 F1 for the baseline). Following the definitions in §4.1, we also experiment with removal of function and content words: both augmented models outperform baseline when all function words are dropped (76.1 F1 for ASR, and 70.2 F1 for MT, and 67.8 F1 for baseline), and ASR augmentation helps when all content words are dropped (68.6 F1 vs. 66.0 F1 for baseline). Finally, we replace one randomly sampled named entity (of type LOC, ORG, or PER) per question with a placeholder, and the performance of ASR-augmented BERT drops less than that of the baseline BERT (by 2.3% and 3.2% respectively). This analysis suggests that ASR augmentation can make models more robust to errors in punctuation, named entities, and content words, and both ASR and MT could help with function word errors.

On the utility of synthetic challenge sets: We advocate that dataset designers always obtain natural data (with natural noise) when possible. However, in the circumstances where collecting natural data is difficult, synthetic data can be useful when reasonably constructed. While the distribution of errors in our synthetically generated challenge sets differs from that in the natural ones (Table 3), we find that the model performance ranking is consistent across all types of noise (Table 4), showing that synthetic noise sets could act as a proxy for model selection. Moreover, augmenting training data with synthetic noise improves model robustness to natural noise for all noise types in this study (Table 6), suggesting that synthetic noise generators may be capturing some aspects of natural noise. Our proposed generators could serve as templates for synthesizing interface noise when collecting natural data is infeasible, but individual practitioners should carefully identify and simulate the likely sources of error appropriate for their applications.

5 Related Work

Question Answering QA systems have a rich history in NLP, with early successes in domain-specific applications (Green et al., 1961; Woods, 1977; Wilensky et al., 1988; Hirschman and Gaizauskas, 2001). Considerable research effort has been devoted to collecting datasets to support a wider variety of applications (Quaresma and Pimenta Rodrigues, 2005; Monroy et al., 2009; Feng et al., 2015; Liu et al., 2015; Nguyen, 2019; Jin et al., 2019) and improving model performance on them (Lally et al., 2017; Wang et al., 2018; Yu et al., 2018; Yang et al., 2019). We too focus on QA systems but center the utility to users rather than new applications or techniques.

There has also been interest in studying the interaction between speech and QA systems. Lee et al. (2018a) examine transcription errors for Chinese QA, and Lee et al. (2018b) propose Spoken SQuAD, with spoken contexts and text-based questions, but they address a fundamentally different use case of searching through speech. Closest to our work is that of Peskov et al. (2019), which studies mitigating ASR errors in QA, assuming white-box access to the ASR systems. Most such work automatically generates and transcribes speech using TTS–ASR pipelines, similar to how our synthetic set is constructed. However, our results show that TTS does not realistically replicate human voice variation. Besides, stakeholders relying on commercial transcription services will not have white-

box access to ASR; our post-hoc mitigation strategies would be better suited for such cases.

Challenge sets Model robustness evaluation with adversarial schemes is common in NLP tasks (Smith, 2012), including dependency parsing (Rimell et al., 2009), information extraction (Schneider et al., 2017), natural language inference (Marelli et al., 2014; Naik et al., 2018; Glockner et al., 2018), machine translation (Isabelle et al., 2017; Belinkov and Bisk, 2018; Bawden et al., 2018; Burlot and Yvon, 2017) and QA (Jia and Liang, 2017; Aspillaga et al., 2020). Unlike most prior work, we do not create our challenge sets to break QA systems, but rather for a more realistic evaluation of the systems' real-world utility.

6 Conclusion

In this work, we advocate for QA evaluations that reflect challenges associated with real-world use. In particular, we focus on questions that are written in another language, spoken, or typed, and the noise introduced into them by the corresponding interface (machine translation, speech recognition, or keyboard). We analyze the effect of synthetic and natural noise in each interface and find that these errors can be diverse, nuanced, and challenging for traditional QA systems. Although we present an initial exploration of mitigation strategies, our primary contribution lies not in the specific challenge sets we construct or in developing new algorithms, but rather in identifying and describing one class of problems that practical QA systems must consider and providing a framework to measure them. We hope insights derived from our study stimulate research in making QA systems ready to face real-world users. We emphasize three considerations:

Sources of error: This work studies errors introduced at the interface stage of QA pipelines. These errors are nearly ubiquitous, as users always interact with QA systems through some kind of interface. Thus, it is important for QA system designers to be mindful of distortions those might introduce. Our analysis can be extended to study the impact of interface-specific factors: for example, how errors vary by keyboard layout (e.g. QWERTY vs. Dvorak or language-specific layouts like AZERTY) or preferred way of typing (e.g. using physical keyboards vs. swipe typing). Another fruitful area of study could lie in examining the accumulated impact of errors resulting from interface combinations (e.g. machine translation of ASR-transcribed queries) and the effects of such interface noise in languages other than English. However, interface distortion represents only one source of error that occurs in practical deployment, and future research would study further sources of variation such as how users may adapt their questions according to the interface used.

Context-driven evaluation: This work focuses on practical evaluation of QA systems that takes into account the challenges associated with their real-world deployment. We hope to encourage development of future user-centered or participatory design approaches to building QA datasets and evaluations, where practitioners work with potential users to understand user requirements and the contexts in which systems are used in practice.

Community priorities for QA systems: While leaderboards on established benchmarks have facilitated rapid progress (Rajpurkar et al., 2016, 2018) and bolstered development of a variety of semantic models (Xiong et al., 2018; Liu et al., 2018; Huang et al., 2018; Devlin et al., 2019), we call for practitioners to consider the orthogonal direction of *system utility* in their model design. We believe these subareas to be complementary, and community attention towards both will help produce NLP systems that are both accurate and *usable*.

Acknowledgments

We thank Aakanksha Naik, Sujeath Pareddy, Taylor Berg-Kirkpatrick, and Matthew Gormley for helpful discussion and the anonymous reviewers for their valuable feedback. This work used the Bridges system, which is supported by NSF award number ACI-1445606, at the Pittsburgh Supercomputing Center (PSC). This research was supported in part by grants from the National Science Foundation Secure and Trustworthy Computing program (CNS-1330596, CNS-15-13957, CNS-1801316, CNS-1914486) and the DARPA KAIROS program from the Air Force Research Laboratory under agreement number FA8750-19-2-0200. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

References

Asma Ben Abacha and Dina Demner-Fushman. 2016. Recognizing question entailment for medical question answering. In *AMIA Annual Symposium Proceedings*, volume 2016, page 310. American Medical Informatics Association.

Francesca Alloatti, Luigi Di Caro, and Gianpiero Sportelli. 2019. Real life application of a question answering system using BERT language model. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 250–253, Stockholm, Sweden. Association for Computational Linguistics.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Akari Asai and Eunsol Choi. 2020. Challenges in information seeking qa: Unanswerable questions and paragraph retrieval. *arXiv preprint arXiv:2010.11915*.

Carlos Aspillaga, Andrés Carvallo, and Vladimir Araujo. 2020. Stress test evaluation of transformer-based models in natural language understanding tasks. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1882–1894, Marseille, France. European Language Resources Association.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint*.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Franck Burlot and François Yvon. 2017. Evaluating the morphological competence of machine translation systems. In *Proceedings of the Second Conference on Machine Translation*, pages 43–55, Copenhagen, Denmark. Association for Computational Linguistics.

Nathan Congdon, Benita O'Colmain, Caroline C. W. Klaver, Ronald Klein, Beatriz Muñoz, David S. Friedman, John Kempen, Hugh R. Taylor, and Paul Mitchell. 2004. Causes and prevalence of visual impairment among adults in the United States. *Archives of Ophthalmology (Chicago, Ill.: 1960)*, 122(4):477–485.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the atis task: The atis-3 corpus. In *HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark. Association for Computational Linguistics.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig, editors. 2020. *Ethnologue: Languages of the World*, 23 edition. SIL International, Dallas, Texas.

Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 813–820. IEEE.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*,

pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Bert F. Green, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: An automatic question-answerer. In *Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference*, IRE-AIEE-ACM '61 (Western), page 219–224, New York, NY, USA. Association for Computing Machinery.

Haihong Guo, Xu Na, and Jiao Li. 2018. Qcorp: an annotated classification corpus of Chinese health questions. *BMC medical informatics and decision making*, 18(1):16.

Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G. Shin, and Karl Aberer. 2018. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 531–548, Baltimore, MD. USENIX Association.

Drew Harwell. 2018. The accent gap. *The Washington Post*.

Tomoki Hayashi, Ryuichi Yamamoto, Katsuki Inoue, Takenori Yoshimura, Shinji Watanabe, Tomoki Toda, Kazuya Takeda, Yu Zhang, and Xu Tan. 2020. ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7654–7658. IEEE.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. DuReader: a Chinese machine reading comprehension dataset from real-world applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 37–46, Melbourne, Australia. Association for Computational Linguistics.

Lynette Hirschman and Robert Gaizauskas. 2001. Natural language question answering: The view from here. *Natural Language Engineering*, 7(4):275–300.

Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen. 2018. Fusionnet: Fusing via fully-aware attention with application to machine comprehension. In *International Conference on Learning Representations*.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496, Copenhagen, Denmark. Association for Computational Linguistics.

Keith Ito and Linda Johnson. 2017. The LJ Speech dataset. `keithito.com/LJ-Speech-Dataset/`.

Pierre Jacquemart and Pierre Zweigenbaum. 2003. Towards a medical question-answering system: a feasibility study. *Studies in health technology and informatics*, 95:463.

Mike Jeffs. 2018. OK Google, Siri, Alexa, Cortana; can you tell me some stats on voice search. Accessed 2020-09-27.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Halil Kilicoglu, Asma Ben Abacha, Yassine Mrabet, Sonya E Shooshan, Laritza Rodriguez, Kate Masterton, and Dina Demner-Fushman. 2018. Semantic annotation of consumer health questions. *BMC bioinformatics*, 19(1):34.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Adam Lally, Sugato Bagchi, Michael A. Barborak, David W. Buchanan, Jennifer Chu-Carroll, David A. Ferrucci, Michael R. Glass, Aditya Kalyanpur, Erik T. Mueller, J. William Murdock, Siddharth Patwardhan, and John M. Prager. 2017. WatsonPaths: scenario-based question answering and inference over unstructured information. *AI Magazine*, 38(2):59–76.

Chia-Hsuan Lee, Shang-Ming Wang, Huan-Cheng Chang, and Hung-Yi Lee. 2018a. ODSQA: Open-domain spoken question answering dataset. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 949–956. IEEE.

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.

Xinjian Li, Zhong Zhou, Siddharth Dalmia, Alan W. Black, and Florian Metze. 2019. SANTLR: Speech annotation toolkit for low resource languages. In *Proc. Interspeech 2019*, pages 3681–3682.

Lanna Lima, Vasco Furtado, Elizabeth Furtado, and Virgilio Almeida. 2019. Empirical analysis of bias in voice-based personal assistants. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 533–538, New York, NY, USA. Association for Computing Machinery.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2018. Stochastic answer networks for machine reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1694–1704, Melbourne, Australia. Association for Computational Linguistics.

Yi-Hung Liu, Yen-Liang Chen, and Wu-Liang Ho. 2015. Predicting associated statutes for legal problems. *Information Processing & Management*, 51(1):194–211.

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*.

Jake Luo, Guo-Qiang Zhang, Susan Wentz, Licong Cui, and Rong Xu. 2015. SimQ: real-time retrieval of similar consumer health questions. *Journal of medical Internet research*, 17(2):e43.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 216–223, Reykjavik, Iceland. European Languages Resources Association (ELRA).

Yajie Miao, Lili Zhao, Chunping Li, and Jie Tang. 2010. Automatically grouping questions in Yahoo! Answers. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 350–357. IEEE.

Alfredo Monroy, Hiram Calvo, and Alexander Gelbukh. 2009. NLP for shallow question answering of legal documents using graphs. In *Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '09, page 498–508, Berlin, Heidelberg. Springer-Verlag.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Vincent Nguyen. 2019. Question answering in the biomedical domain. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 54–63, Florence, Italy. Association for Computational Linguistics.

Joe Osborne. 2016. Why 100 million monthly Cortana users on Windows 10 is a big deal. *TechRadar*. Accessed 2020-09-27.

Adam Palanica, Anirudh Thommandram, Andrew Lee, Michael Li, and Yan Fossat. 2019. Do you understand the words that are comin outta my mouth? Voice assistant comprehension of medication names. *NPJ digital medicine*, 2(1):1–6.

Vijayaditya Peddinti, Guoguo Chen, Vimal Manohar, Tom Ko, Daniel Povey, and Sanjeev Khudanpur. 2015. JHU ASpIRE system: Robust LVCSR with TDNNS, iVector adaptation and RNN-LMS. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 539–546. IEEE.

Denis Peskov, Joe Barrow, Pedro Rodriguez, Graham Neubig, and Jordan Boyd-Graber. 2019. Mitigating noisy inputs for question answering. In *Proc. Interspeech 2019*, pages 789–793.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel

Vesely. 2011. The Kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society.

Alisha Pradhan, Kanika Mehta, and Leah Findlater. 2018. "Accessibility came by accident": Use of voice-controlled intelligent personal assistants by people with disabilities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–13, New York, NY, USA. Association for Computing Machinery.

Paulo Quaresma and Irene Pimenta Rodrigues. 2005. A question answer system for legal information retrieval. In *Proceedings of the 2005 Conference on Legal Knowledge and Information Systems: JURIX 2005: The Eighteenth Annual Conference*, page 91–100, NLD. IOS Press.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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

Antoine Raux, Brian Langner, Dan Bohus, Alan W Black, and Maxine Eskenazi. 2005. Let's go public! Taking a spoken dialog system to the real world. In *Proc. Interspeech 2005*.

Abhilasha Ravichander, Alan W Black, Shomir Wilson, Thomas Norton, and Norman Sadeh. 2019. Question answering for privacy policies: Combining computational and legal perspectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4947–4958, Hong Kong, China. Association for Computational Linguistics.

Laura Rimell, Stephen Clark, and Mark Steedman. 2009. Unbounded dependency recovery for parser evaluation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 813–821, Singapore. Association for Computational Linguistics.

Kanthashree Mysore Sathyendra, Abhilasha Ravichander, Peter Garth Story, Alan W. Black, and Norman Sadeh. 2017. Helping users understand privacy notices with automated query answering functionality: An exploratory study. Technical report, Carnegie Mellon University.

Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A. Gers, and Alexander Löser. 2017. Analysing errors of open information extraction systems. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 11–18, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

Noah A. Smith. 2012. Adversarial evaluation for models of natural language. *arXiv preprint*.

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2017. Listening while speaking: Speech chain by deep learning. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 301–308. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018. R^3: Reinforced ranker-reader for open-domain question answering. In *AAAI Conference on Artificial Intelligence*.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. ESPnet: End-to-end speech processing toolkit. In *Proc. Interspeech 2018*, pages 2207–2211.

Robert Wilensky, David N. Chin, Marc Luria, James Martin, James Mayfield, and Dekai Wu. 1988. The berkeley unix consultant project. *Computational Linguistics*, 14(4).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *arXiv preprint*.

William A. Woods. 1977. Lunar rocks in natural English: Explorations in natural language question answering. In *Linguistic Structures Processing*, pages 521–569. North Holland, Amsterdam.

Caiming Xiong, Victor Zhong, and Richard Socher. 2018. DCN+: Mixed objective and deep residual coattention for question answering. In *International Conference on Learning Representations*.

Min Yang. 2015. Deep Markov neural network for sequential data classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 32–37, Beijing, China. Association for Computational Linguistics.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Adams Wei Yu, David Dohan, Quoc Le, Thang Luong, Rui Zhao, and Kai Chen. 2018. Fast and accurate reading comprehension by combining self-attention and convolution. In *International Conference on Learning Representations*.

Yanli Zheng, Richard Sproat, Liang Gu, Izhak Shafran, Haolang Zhou, Yi Su, Daniel Jurafsky, Rebecca Starr, and Su-Youn Yoon. 2005. Accent detection and speech recognition for Shanghai-accented Mandarin. In *Proc. Interspeech 2005*.

A Reproducibility details of models

We use the pre-trained AllenNLP implementations of BiDAF and BiDAF-ELMo[18] (Gardner et al., 2018) and the HuggingFace implementation of BERT.[19] We fine-tune BERT and RoBERTa on SQuAD with a learning rate of $3e{-}5$ for 2 epochs, with a maximum sequence length of 384. All models achieve good performance on the SQuAD dataset. Our trained models achieve the following F1 scores on SQuAD development set: BiDAF: 77.82, BiDAF-ELMo: 80.68, BERT: 88.75, RoBERTa: 89.93.

B Keyboard noise in the wild

Common examples of keyboard typos include replacing a character with the one corresponding to an adjacent key *(frame→framd)*, inserting or deleting characters *(between→betwen, agency→agenchy)*, and swapping adjacent characters within words *(beroids→beriods)*. Such errors exist even in textual QA datasets collected in relatively controlled settings: for example, all the error examples above actually occur in SQuAD. In a real-life situation of information need, where the user produces the question without being exposed to the context and the answer, these errors will likely be even more pervasive. We qualitatively analyze a sample from a dataset of questions collected from the Yahoo! Answers platform (Miao et al., 2010), randomly selecting 50 questions from each topic (Science, Internet, and Hardware). We manually identify non-standard spellings and discard ones that are intentional, such as slang *(thanks→thanx)* or expression of emotion *(so→sooo)*. Since we are specifically interested in the errors that happen in the process of typing, we also separate out errors that could have originated in the user's mind; for example, the most frequent class of errors is omission or insertion of apostrophes in contractions, possessives and plurals, but all of them could plausibly be explained by the user's intention. Other common error types we find are incorrect whitespace placement and character substitutions (mostly plausible human errors), and character insertions, deletions or swapping adjacent characters within words (mostly interface errors); statistics and error examples can be found in Table 7.

[18] github.com/allenai/allennlp-hub

[19] github.com/huggingface/pytorch-transformers

Error type	Examples	#Errors
Apostrophe	*it's→its, devices→device's*	55 (0)
Whitespace	*anyone→any one, a lot→alot*	18 (4)
Deletion	*too→to, school→schol*	18 (10)
Substitution	*warranty→warrenty, will→well*	12 (2)
AdjSwap	*type→tpye, piece→peice*	11 (9)
Insertion	*answer→asnswer, lose→loose*	9 (5)
KeySwap	*of→if*	1 (1)

Table 7: Examples of common error types observed in a manual analysis of the Yahoo! Answers questions. Examples identified as interface errors are highlighted in blue. #Errors is total number of typographical errors, with # of interface errors in parentheses.

C Filtering interface misspellings

Our source of human keyboard errors is the Wikipedia list of common English misspellings; some of them are likely to occur in the process of typing (e.g. *and→adn*), while others can plausibly be explained by user misconception (e.g. *recieve→receive*). Since our work focuses on interface errors specifically, we would like to only retain errors from the former category.

Our filtering approach is based on two assumptions: (a) interface errors must be plausible under the keyboard layout, and (b) misspellings that preserve pronunciation of the original word (e.g. *article→artical*) are more likely to be non-interface errors coming from users themselves. We use a two-step filtering heuristic: first, we retain only error categories likely to be explained by the interface noise (character deletion and insertion, adjacent character swap or adjacent key swap in QWERTY layout), and then discard spellings with similar pronunciations. Pronunciations are obtained via the Epitran G2P system (Mortensen et al., 2018), and similarity is determined by weighted edit distance.

On a sample of 100 Wikipedia misspellings manually labeled as interface or non-interface errors, the proposed heuristic shows 83% agreement with human annotation. Applying the heuristic to the initial 4,518 word–spelling pairs, we obtain a set of 1,742 interface errors for 1,489 English words.

D Voice variation in ASR

This section describes the details of the voice variation experiments discussed in §4.1. The numbers used to generate Figures 1a and 1b are presented in Tables 8 and 10 respectively.

Synthetic variation We generate the synthetic voices using Google English Text-to-Speech system with four different accent settings (Australian, British, Indian, and US) and two gender settings (male and female voices). The performance of all models on these voices is presented in Table 8. All QA models achieve highest F1 score when the questions are voiced with a US accent, which is likely explained by the ASR component being optimized for this accent specifically. Neither gender setting consistently leads to best performance across all models and accents. BiDAF and RoBERTa achieve highest scores when the US female synthetic voice is used, and BiDAF-ELMo and BERT perform best with the US male synthetic voice.

Natural variation We record the spoken versions of the 1,190 XQuAD questions voiced by three human annotators: H1 (Indian female), H2 (Russian female), and H3 (Indian male). The same three annotators and an additionally recruited annotator H4 (Scottish male) also voiced the same random sample of 100 XQuAD questions to measure the effect of voice variation in content-controlled setting. The summary statistics (mean and standard deviation) for the sample of speakers are shown in Figure 1b, and the breakdown of each model's score by speaker is presented in Table 10. To collect a set of recordings that is more representative of the real-life use cases, we do not control for recording conditions and other confounds, so our per-speaker results alone are not meant to be taken as evidence of the ASR or QA models being better-tuned for any of the mentioned demographics.

E Input language variation in MT

Table 9 presents the results of the query language variation experiment (§4.1, Figure 2). In this experiment, we use XQuAD human translation of questions into ten languages as inputs, translating them back into English through the Google Translation API. The table also reports the results on the original English SQuAD questions to serve as a skyline. As expected, lower-resource languages and languages that are more typologically divergent from English (the QA system's language) pose the biggest challenge for the MT–QA pipeline.

F Robustness experiments

Table 11 presents the question repair and data augmentation results on both synthetic and natural noise for all interfaces. Synthetic noise sets were used for development and tuning in all experiments. Table 11 also breaks down data augmentation results by the specific augmentation noise source. Training on ASR noise proves helpful for natural keyboard noise as well as natural ASR noise, and robustness to natural translation noise is only improved by augmenting the data with its synthetic counterpart.

G ASR system benchmarking

To benchmark both the ESPnet CommonVoice ASR system, which we use for data augmentation, and the Google ASR, which was used to create ASR challenge sets from recorded XQuAD questions, we also transcribe the natural and synthetic challenge set recordings with ESPnet ASR. ESPnet achieves 56.8% and 70.1% WER for synthetic and natural voices respectively, while Google ASR gets a WER of 16.6% and 30.7% respectively (Table 3).

H Numeral handling and ASR interfaces

Correctly transcribing numerals is often important for producing a correct answer in an ASR–QA pipeline. Even a different representation of the same quantity in the question and in the context passage creates additional difficulties for the QA system. To additionally analyze the effect of handling numerals in ASR engines, we combine BERT with Kaldi (Povey et al., 2011) or Google speech recognizers and compare their performance on the portion of XQuAD questions containing numerals (XQuAD-NUMBERS) and the remaining questions (XQuAD-NONUM). With the questions narrated by human annotators, the QA pipeline performs worse on XQuAD-NUMBERS than XQuAD-NONUM with either Kaldi (38.39 F1 and 44.30 F1 respectively) or Google ASR (64.44 F1 and 70.86 F1 respectively). In case of Kaldi, we hypothesize that the discrepancy might be partially explained by the speech recognizer outputting numbers in their spelled-out form rather than numeric form. To test this hypothesis, we convert all numerals in the original written XQuAD-NUMBERS questions into their spelled-out form and observe a drop in performance from 87.10 F1 to 82.88 F1 on this subset. However, the representation mismatch is only one of many challenges: unlike Kaldi, Google ASR outputs numerals as digits, but the corresponding pipeline still shows worse performance on spoken XQuAD-NUMBERS.

Model	AU		GB		IN		US	
	Female	Male	Female	Male	Female	Male	Female	Male
BiDAF (Seo et al., 2017)	64.14	64.76	60.45	63.73	64.09	64.80	65.93	66.39
BiDAF-ELMo (Peters et al., 2018)	67.84	67.49	65.08	67.04	68.13	68.94	70.50	70.30
BERT (Devlin et al., 2019)	74.54	73.87	70.56	72.79	73.65	74.47	77.42	77.02
RoBERTa (Liu et al., 2019)	78.86	78.79	76.37	78.27	78.83	80.13	81.11	81.38

Table 8: Performance of different QA models in the TTS-ASR pipeline with different synthetic voices. We use Google Text-to-Speech with different accent and gender settings, and Google Speech-to-Text optimized for English–US as the speech recognizer.

Model	en	es	hi	vi	de	ar	zh	el	ru	th	tr
BERT	84.66	79.86	76.75	77.14	79.98	75.45	76.39	76.96	78.06	71.03	76.98
RoBERTa	84.42	81.65	79.61	78.77	82.13	76.41	78.88	79.6	79.67	74.68	79.28

Table 9: QA performance on XQuAD human translations of SQuAD questions in different source languages posed to an English QA system. Questions in each non-English language are translated to English using the Google MT system, and the performance on the original English questions is reported as a skyline.

Model	H1	H2	H3	H4
BiDAF	58.14	62.86	31.60	60.07
BiDAF-ELMo	56.15	62.65	29.30	62.48
BERT	59.77	67.27	32.98	65.63
RoBERTa	60.74	74.31	34.14	67.48

Table 10: Performance of the different QA models on different human annotator voices: Indian Female (H1), Russian Female (H2), Indian Male (H3), and Scottish Male (H4). We do not control for recording conditions and other confounds in this experiment, so our results are not meant to act as evidence of ASR systems being more effective for any particular demographic.

BERT Model	$XQuAD_{EN}$		ASR		MT		Keyboard	
	EM	F1	EM	F1	EM	F1	EM	F1
Synthetic								
BERT	**72.77**	**84.66**	61.93	77.02	**67.23**	79.08	61.68	74.43
+ NE Repair	72.94	84.78	62.10	77.23	67.31	79.19	63.78	75.31
+ Content Repair	72.94	84.77	62.02	77.12	67.31	79.14	62.61	74.34
+ Spelling Augmentation	72.35	83.89	56.81	73.68	65.63	78.09	**67.31**	**78.83**
+ ASR Augmentation	71.93	83.41	**66.13**	78.29	66.13	78.29	65.46	76.65
+ Translation Augmentation	70.76	83.17	61.09	76.42	66.72	**79.70**	59.83	72.29
+ Spelling+ASR+Translation Augmentation	67.48	80.64	**66.13**	**79.82**	64.20	77.18	64.28	77.63
Natural								
BERT	72.77	84.66	52.94	67.13	68.82	79.98	69.16	81.84
+ NE repair	**72.94**	**84.78**	53.03	67.34	68.82	80.05	69.58	82.22
+ Content repair	**72.94**	84.77	52.61	67.01	68.32	79.76	70.25	**82.60**
+ Spelling Augmentation	72.35	83.89	50.84	66.04	68.49	80.20	70.25	82.22
+ ASR Augmentation	71.93	83.41	**64.37**	**75.89**	68.65	80.32	**70.76**	82.43
+ Translation Augmentation	70.76	83.17	53.70	68.11	**68.90**	**80.83**	68.57	81.05
+ Spelling+ASR+Translation Augmentation	67.48	80.64	62.02	74.61	66.81	80.25	65.88	78.55

Table 11: Effect of question repair and data augmentation on BERT performance on both synthetic and natural noise for the three interface types. Data augmentation results are presented separately for each source of training synthetic noise. Synthetic noise sets are used for development and tuning in all experiments.