GameQA: Gamified Mobile App Platform for Building Multiple-Domain Question-Answering Datasets

Njáll Skarphéðinsson[1,2], **Breki Guðmundsson**[2], **Steinar Þ. Smári**[1], **Marta K. Lárusdóttir**[1], **Hafsteinn Einarsson**[3], **Abuzar Khan**[2], **Eric Nyberg**[2], **Hrafn Loftsson**[1]
[1] Department of Computer Science, Reykjavik University, Reykjavik, Iceland
[2] Language Technologies Institute, Carnegie Mellon University, PA, USA
[3] Department of Computer Science, University of Iceland, Reykjavik, Iceland
{nskarphe, abuzark, en09}@andrew.cmu.edu, hafsteinne@hi.is
{brekig18, steinars21, marta, hrafn}@ru.is

Abstract

The methods used to create many of the well-known Question-Answering (QA) datasets are hard to replicate for low-resource languages. A commonality amongst these methods is hiring annotators to source answers from the internet by querying a single answer source, such as Wikipedia. Applying these methods for low-resource languages can be problematic since there is no single large answer source for these languages. Consequently, this can result in a high ratio of unanswered questions, since the amount of information in any single source is limited. To address this problem, we developed a novel crowd-sourcing platform to gather multiple-domain QA data for low-resource languages. Our platform, which consists of a mobile app and a web API, gamifies the data collection process. We successfully released the app for Icelandic (a low-resource language with about 350,000 native speakers) to build a dataset which rivals large QA datasets for high-resource languages both in terms of size and ratio of answered questions. We have made the platform open source with instructions on how to localize and deploy it to gather data for other low-resource languages.

1 Introduction

Replicating well known Question-Answering (QA) data collection methods, such as those used to create the SQuAD (Rajpurkar et al., 2016) and TyDi (Clark et al., 2020) datasets, for low-resource languages poses a few problems. First, many large-scale QA datasets are gathered using a single source for answers, e.g. Wikipedia. This is problematic since low-resource languages do not have access to any single, large knowledge base from which information can be extracted to create such a dataset. Second, QA datasets gathered in an information-seeking manner (where the question is asked prior to finding the answer), will have questions that cannot be answered by the answer source(s). As we will show, the ratio of answerable questions is positively correlated with the amount of content in an answer source. Third, many of these methods rely on paid workers to perform the laborious task of annotating data and the necessary funds may not be available in regions of low-resource languages.

In this paper, we introduce GameQA, a crowd-sourcing platform to build QA datasets. GameQA consists of a mobile trivia app (for iOS and Android) and a web API. GameQA, which is open source, is specifically designed to gather QA data for low-resource languages. It can be trivially localized and published for specific geographical regions. The main contributions of GameQA are as follows:

- **Gamification:** It incorporates numerous aspects of gamification to increase the number of annotations provided per user. This includes rewarding points, level-ups, streaks, avatar upgrades, and prestige tokens to users as they contribute to the data collection.
- **Social features:** The users are made aware of their contributions relative to other users. This includes a leaderboard and notifying users once another user has answered their question.
- **Cultural relevance:** Our results show that GameQA gathers questions which are relevant to the culture, history, and geography of the region in which it is employed.
- **Multiple Answer Sources:** The platform's API integrates Google's Programmable Search Engine[1] to allow users to find answers

[1] https://developers.google.com/custom-search/v1/introduction

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics
System Demonstrations, pages 152–160

on multiple websites, thus seamlessly constructing a multiple-domain QA dataset.

To spur QA research for low-resource languages, we have made the GameQA platform open source with instructions on how to localize it and subsequently release it for any geographical area[2].

2 Related Work

In recent years, the literature has seen an explosion in the number and diversity of QA datasets (Cambazoglu et al., 2021). The most prevalent type of QA datasets are sentence classification and span-prediction datasets. These include documents, questions, and demarcated answer spans that a machine learning model must learn to predict for a given question. Rajpurkar et al. (2016) introduced SQuAD, one of the first large span-prediction datasets. They crowd-sourced the creation of the dataset by showing crowd-workers an excerpt from Wikipedia and tasking them with writing a question whose answer is contained within the excerpt. This results in QA data with a high lexical overlap between questions and answer paragraphs (Ribeiro et al., 2019; Gan and Ng, 2019) which can lead to biased data (Shinoda et al., 2021). In this paper, we will refer to span-prediction datasets constructed in this manner as being *squad-like*. CoQa (Reddy et al., 2019), NarrativeQA (Kočiský et al., 2018), and NewsQA (Trischler et al., 2016) are other examples of datasets falling into this category.

To address the problems associated with squad-like datasets, researchers have developed ways that encourage information-seeking behavior during question elicitation. The aim is to emulate human curiosity by having annotators ask questions about something they do not know the answer to. WikiQA (Yang et al., 2015), which poses the problem as a sentence classification problem instead of span prediction, is orders of magnitude smaller than SQuAD (3,047 vs. 100,000 questions, respectively). However, WikiQA brings forth interesting ideas, such as collecting QA data in an information-seeking manner and using web-search queries as a means to capture the curiosity of information-seeking users. This web-search query-based approach was later adopted by larger information-seeking QA datasets, such as MS-Marco (Bajaj et al., 2016), SearchQA (Dunn et al., 2017), and Natural Questions (Kwiatkowski et al., 2019). TyDi is an example of an information-seeking dataset constructed using answer paragraphs from the encyclopedic domain (Clark et al., 2020). Clark et al. showed crowd-workers a paragraph from Wikipedia, but instructed them to ask a question that was not answerable by the paragraph.

Just as QA datasets can differ in terms of how they source their questions (e.g., squad-like or information-seeking), they can also be categorized in terms of where their answer paragraphs are sourced, i.e. the domain that contains the answers. A very common practice is to constrain a dataset to a single domain – this is the case for the majority of over 80 QA datasets reviewed by Rogers et al. (2020). SQuAD, WikiQA, TyDi, and Natural Questions are examples of such single-source datasets, i.e. they all source answer paragraphs from Wikipedia only. Examples of datasets sourcing answers from another notable domain, the news domain, are NewsQA and CNN/Daily Mail (Hermann et al., 2015). However, low-resource languages are unlikely to have access to a single source that contains enough information to construct a large QA dataset.

Multiple-domain QA datasets have also been created. MS-Marco, which utilized Bing[3] search queries, used a proprietary state-of-the-art passage retrieval system at Bing to match queries with answer paragraphs on the internet. Since MS-Marco relies on such an algorithm, replicating their methods (i.e. for other languages) is impossible. MMQA, a multiple-domain, squad-like dataset in English and Hindi, was created by web-crawling and subsequently having annotators write questions about the crawled articles (Gupta et al., 2018). However, it is likely that it suffers from the same problems as other squad-like datasets. To the best of our knowledge, there exists no easily reproducible method in the literature to gather a multi-domain dataset where the questions reflect information-seeking intent.

2.1 QA for Icelandic

In the last few years, Icelandic has been growing considerably with regard to language resources (Nikulásdóttir et al., 2022). However, for many natural language processing tasks it still lacks the necessary resources. For reading comprehension and open QA tasks, there only exists one dataset

[2] https://github.com/cadia-lvl/GameQA

[3] https://www.bing.com

for Icelandic (Snæbjarnarson and Einarsson, 2022). It was created using the same information-seeking process as was introduced with TyDi. Furthermore, the authors specifically mentioned that they exhausted the Icelandic Wikipedia[4] when creating questions for the dataset, thereby highlighting the need to include more domains both for question elicitation and answer annotation.

3 The GameQA Platform

Our crowd-sourcing platform consists of a mobile app and web API. The mobile app was written in React Native, the web server in Node.js, and the underlying database is MongoDB.

We recruited users by sending an email to all students at Reykjavik University and by advertising the app on social media platforms. The app was distributed through Apple App Store (iOS version) and Google Play Store (Android version), and, in both cases, only made accessible in Iceland.

The users form a community where they help each other finding answers to user generated questions. For example, the app might ask a user to write a question. Later on, another user would be tasked with reviewing it. Once it passes peer review, a third user would be tasked with finding a specific paragraph on a web page containing the answer, using an integrated web-search interface. Lastly, another user would verify the answer. These tasks are served randomly and users are not able to review their own content.

3.1 User centered development

In the design and development of GameQA, we applied user-centred design methodology through iterative development and three prototypes. In the first iteration, a web-based interface was evaluated. As a result, in the second iteration, the interface was simplified and gamification was added. When evaluating the second version, users pointed out the need for a mobile phone interface. In the final iteration, the mobile app was thus developed and evaluated. Involving users in the design and the development of the application improved the final result and the user experience. By qualitatively analysing the user interface prior to launch, we were able to understand which gamification features could increase adoption and usage of the app.

[4]At the time, the Icelandic Wikipedia had only 3,730 pages with more than 250 characters.

4 Gamification and Social Features

Gathering and annotating QA data is a laborious and repetitive task. Since the crowd-workers of GameQA are not financially compensated for their contributions, and thus have little incentive to participate, GameQA leverages gamification to incentivize the users and to give them positive feedback when they contribute to the data collection.

4.1 User levels and avatars

Every user collects points by completing tasks. For each completed task, the user is rewarded with 1 point. Upon completing a certain number of tasks, the user is awarded with a "level-up". We used an ad-hoc formula (see Equation 1) to calculate the number of tasks T in order to complete a specific level L:

$$T_L = \lfloor 2.5 \times L^{1.1} \rfloor \quad (1)$$

Here $T_L \in \mathbb{N}, \forall L \in \mathbb{N}$. For example, a user would have to complete $\lfloor 2.5 \times 1^{1.1} \rfloor = 2$ tasks for the first "level-up", and $\sum_{l=1}^{20} T_l = 667$ tasks to complete all 20 levels.

Users are also given avatars which change as the users level up. Since users can see each other's avatars, they are a signal from a user to the community about their status.

4.2 Prestige tokens

Once we rolled out the data collection, we were doubtful that any user would finish the 667 tasks required to complete all 20 levels. After the first day, however, we realized that a few completed 667 tasks within 24 hours and, subsequently, stopped playing. We hypothesized that this was because users had little motivation to continue annotating once they had reached the maximum level. As a result, taking inspiration from gaming franchises like `Call of Duty`[5], we added *Prestige Tokens*. The prestige tokens work as follows: Users are prompted when they finish level 20 to restart the game at level 1, but with a token that appears next to their avatar which signals to other users that they have finished the game once over. The prestige tokens then change color, every time the user reaches level 20.

4.3 Leaderboard

We implemented a live leaderboard within GameQA which allows users to see how they are

[5]`https://www.activision.com/`

performing relative to other users. We observed significant competitiveness amongst some users after adding this feature. For example, some users spent several hours per day annotating data, in order to achieve the highest rank. The avatar of the highest ranking user was given a crown to further incentive users to compete for the highest rank.

It is worth mentioning that even though users competed to achieve high ranks they were informed that the data collection was a collaborative effort, for the purpose of compiling a training corpus for Icelandic QA models.

4.4 User notifications

Once a question has been answered by the community, a notification is sent to the author of the question telling them that they can see the answer (and who answered it) in the app. This serves as an important way for users to see that their contribution is impacting the data collection.

5 Data Collection Steps

In total, the data collection consists of five different stages that each QA pair has to pass: 1) question elicitation, 2) question review, 3) web search and answer paragraph selection, 4) answer span marking, and 5) answer review[6]. Tasks are randomly served to users – subsequent tasks are thus independent of one another.

For our Icelandic QA dataset, which we call *RUQuAD* (Reykjavik University Question Answering Dataset)[7], we sourced answers from five sources in four separate domains: The Icelandic Wikipedia[8], "Vísindavefurinn" (The Icelandic Web of Science)[9], the news websites `mbl.is` and `visir.is`, and "Stjórnarráðið" (The Icelandic Government Information website)[10].

5.1 Question elicitation

Users are shown an image and asked to write a question that comes to mind. However, users are not constrained to ask a question about the image itself. Instead, the image serves as a stimulus for curiosity. To gather the set of images, we first constructed a list of 78 broad topics. From there, we found one image related to each topic.

[6]Demonstrated in a YouTube video: https://www.youtube.com/watch?v=PmCR7v_KDhQ

[7]https://github.com/cadia-lvl/RUQuAD

[8]https://is.wikipedia.org/wiki/

[9]https://www.visindavefur.is/

[10]https://www.stjornarradid.is/

5.2 Question review

Since we seek to gather questions based on the users' curiosity with minimal guidance and influence, we purposefully place little restrictions on the nature of the questions. Users are asked to rate questions given the following criteria:

Clarity If it is clear what the author of the question is asking for.

Consistency If the answer is unlikely to change depending on whom or when you ask.

Answer length If it seems like this question could be answered in three sentences or less.

We chose to include the *Answer Length* criteria in order to simplify other annotation tasks such as answer reviews. Each question has to pass all of these criteria in two separate reviews performed by two separate users. Researchers seeking to localize GameQA can modify these criteria if needed.

5.3 Web search and paragraph selection

A distinguishing feature of our data collection is the users' ability to find answers in various different sources and domains instead of only linking a question to a Wikipedia article. When searching for an article online containing an answer to a given question, the users form a search string that they believe will lead to success, i.e. for which an answer will be found (see Figure 1). This is carried out in very much the same way as a user of a search engine performs a web search.

Once the users find a website that contains the information necessary to answer the question, they select the exact paragraph that contains the answer. If annotators are not able to find an answer, they can mark it as *unanswerable*.

5.4 Answer span marking

Once a question has been linked to an answer paragraph, the question and the attached paragraph is shown to users. First, they are asked whether or not they think that the answer is contained within the paragraph. If the user responds in the affirmative, they are then tasked with selecting the first and last word (the span) of the answer (see Figure 2). However, if the question is a YES or NO question, then the user will mark it as such with the right answer.

Figure 1: A screenshot from the mobile app demonstrating the interface for the *Web search and paragraph selection* task. The question (in Icelandic) presented to the user is “Hver drap Frankenstein” (Who killed Frankenstein). The user has formed the search string “Frankenstein”, and a list of search results from the five sources appears below. At the top, the users can see their avatar, level, position on the leaderboard, and their progress towards their next level.

Figure 2: A screenshot from the mobile app demonstrating the interface for the *Answer span marking* task. The question (in Icelandic) presented to the user is “Í hvaða heimsálfu er Perú” (In which continent is Peru). The user has marked “Suður-Ameríku” (South America) as the answer.

5.5 Answer review

The last step in this pipeline is the answer review step. Similar to the question review step, each answer has to pass two separate reviews from two separate users. The review step consists of a single question, asking users if they believe an answer shown to them to be correct or not. The users are not required to know the precise answer to the question, instead they use their reading comprehension skills and judgement to determine if the answer seems correct.

6 Results and Data Analysis

Throughout our QA collection process for Icelandic using GameQA, 1,524 users created an account. Roughly $\frac{2}{3}$ of those (1,024 users) contributed content to the creation of the RUQuAD dataset. By the end, they had generated 23,036 questions, 20,730 (90%) of which passed the double peer review. 12,772 answers were annotated and reviewed, resulting in an unanswerable ratio of 38.4%. A preliminary analysis suggests that approximately 30% of the questions that either failed the peer review or were marked as unanswerable might have been mislabeled as such. As a result, the unanswerable rate might become considerably lower with additional labeling after crowd-sourcing the data.

There is a remarkable diversity in the number of answer articles. 7,835 articles were gathered in total for the 12,722 answers, i.e. 1.64 answers per article. This ratio is roughly 2.05 and 200 for TyDi and SQuAD, respectively. We expect that more diverse answer paragraphs will help a machine learn-

ing model, trained on the data, to generalize better. The distribution of articles over the sources is as follows: 68.3% came from the Icelandic Wikipedia, 18.4% from The Icelandic Web of Science, 13.1% from the two news websites, and 0.2% from the Government Information website.

6.1 Understanding user contributions

Since the crowd-workers weren't paid, but rather users playing a game in their own free time, the strength of each users contribution was mostly impacted by the time they were willing to spend on the app. In total, the users performed 137,972 annotation tasks (elicit questions, review questions, find answers, label answers, review answers). As Figure 3 shows, the amount of work performed per user follows a pareto distribution.

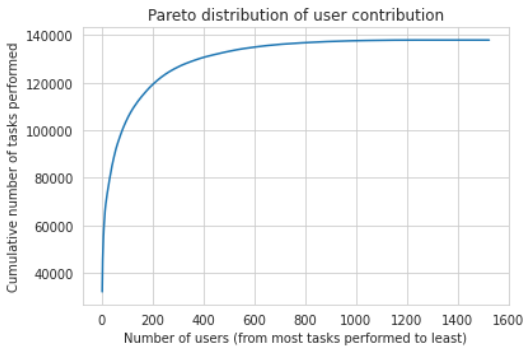

Figure 3: Tasks completed per user follow a pareto distribution where a minority of the 1,524 users contributed a majority of the content.

6.2 Unanswerable Questions

As mentioned in Section 2, a particular problem with information-seeking methods for low-resource languages is the high ratio of unanswerable questions – this can be observed in Table 1.

Icelandic has fewer Wikipedia articles (54,121) than all languages in TyDi. Yet, by leveraging multiple answer sources with GameQA, we achieved an unanswerable rate of 38% which is lower than all languages in TyDi, except Arabic.

6.3 Span length distribution

Out of the five annotation steps, marking answer spans is the step that requires the highest degree of standardization of annotation. Without such standards (or precise guidelines) and a way of enforcing them, the annotators will not mark answer spans in a consistent manner.

Language	Number of Wikipedia articles	Ratio of questions with an answer span
Russian	1,816,916	51%
Japanese	1,324,304	32%
Arabic	1,165,575	69%
Indonesian	620,863	34%
Korean	587,573	22%
Finnish	530,420	41%
Thai	147,378	43%
Bengali	122,041	35%
Telugu	76,259	27%
Kiswahili	71,570	22%

Table 1: A comparison of the ratio of questions, which had an answer span, with the number of Wikipedia articles, for each of the 10 non-English languages in the TyDi dataset. The Pearson correlation coefficient is $p = 0.54$.

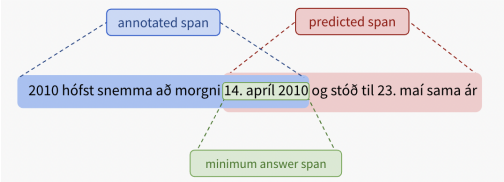

Figure 4: A comparison of the minimum answer span with the annotated and predicted spans. The underlying subsentence is: "hófst snemma að morgni 4. apríl 2010 og stóð til 23. maí sama ár" (started early in the morning on April 4, 2010 and lasted until May 23 of the same year).

Figure 4 shows a comparison of the minimum answer span (needed to answer a question) with the span annotated by a user and the span predicted by the IceBERT model (Snæbjarnarson et al., 2022), fine-tuned on our dataset. We expect this discrepancy between ground truth labels and the prediction to be a result of lack of alignment amongst annotators when marking answer spans. We propose that researchers that localize GameQA standardize and shorten the answer spans where needed, once the crowd-sourcing has concluded.

Table 2 shows summary statistics for the answer span lengths for three different datasets. Higher variance in answer span lengths in RUQuAD is to be expected since enforcing annotation standards across thousands of crowd-workers is non-trivial.

Dataset	Span Length	Standard Deviation
SQuAD	19.75	20.73
TyDi	25.77	46.12
RUQuAD	75.64	91.52

Table 2: Summary statistics of answer span lengths (character count)

6.4 Cost-effective data collection

A clear advantage of localizing GameQA for data collection for other languages is the possibility of gathering QA data in a cost-effective manner. By gamifying the data collection, we were able to create a large-scale QA data set, gathered by thousands of crowd-workers without the need of hiring, training, and managing annotators. The majority of the cost we incurred with GameQA was the cost of developing the platform. By making the code open source, we hope to enable researchers around the world to gather cost-effective large-scale multiple-domain QA data for low-resource languages.

6.5 Cultural relevance

As a result of having thousands of annotators, we observe a considerable diversity in terms of the range of topics users asked about. Furthermore, we notice that our proposed method is able to gather questions which are representative of the local history and culture. In order to build QA systems, researchers for low-resource languages might be tempted to translate large English datasets. However, translation of English datasets will not produce questions relevant to local culture, history, geography, etc. Out of 100 questions from our dataset, sampled uniformly at random, 33 were directly asking about local (Icelandic) culture, history, or geography. This emphasis on culture-related questions can possibly be attributed to some extent to the images used in the prompting step, but it also highlights how the right combination of annotators and prompts can lead to greater culture focus in the resulting data.

7 Conclusion

In this paper, we have presented GameQA – a novel mobile trivia game platform for collecting QA data for low-resource languages. We successfully gamified the experience to increase the number of annotations tasks performed per user and conducted different iterations of user experience testing. The QA data gathered by GameQA's users is culturally relevant for the language and/or geographical region in question. We have made GameQA open source, with instructions on how to localize and subsequently release it for particular geographical areas.

We believe that our platform can help to reduce the cost and time associated with collecting QA data for low-resource languages. Our method opens up new areas of research e.g. comparing different prompting methods, such as image vs. text prompts, as well as possible advancements for QA research in languages where traditional methods might fail to gather a large-scale QA dataset.

Given the success of the gamification for collecting QA data, we propose that gamified crowdsourcing can be leveraged to gather data for other NLP tasks as well. For an app such as GameQA, there is a target user base that is interested in triva and knowledge and thus willing to annotate data in this manner. Similarly, for other NLP tasks, such as machine translation, there exists a potential user base of multilingual persons that are greatly interested in languages and translation. We see great potential in applying the knowledge learned through implementing GameQA for such tasks.

Limitations

The question elicitation part of GameQA is different from prior work. An image is shown to the user instead of a textual prompt to inspire questions. It is unclear what effect that decision has on the chances of the question being answerable since users could make more or less challenging questions when prompted with images instead of text. Furthermore, the source from which the images are taken could further influence answerability. Future work will need to reveal the difference between prompting with text or images.

In GameQA, the user is responsible for finding the article that could contain an answer to a given question. This step was automated in prior work by selecting top search engine results. This approach gives the user more freedom when looking for the answer. Still, it could also limit their ability to find answers since they are responsible for performing Google search queries themselves. Although an answer might exist, their queries might not suffice to identify relevant candidate pages.

Furthermore, it is likely that some of the questions asked are ambiguous, i.e. that for a given

question more than one correct answer is possible. In such cases, a rewrite of the question, for the purpose of clarifying its interpretation, might be beneficial (Min et al., 2020). In GameQA, this would require an additional task in the question review step (see Section 5.2).

Ethics Statement

The data collection process in GameQA consists of collecting paragraphs, from a set of sources/domains (see Section 5), in which answers can be found to given questions. Before starting our RUQuAD corpus collection process, we obtained formal permissions from The Icelandic Web of Science, the news cites `mbl.is` and `visir.is`, and the Icelandic Government Information website, to freely include paragraphs from their sources in our corpus. For the last domain, the Icelandic Wikipedia, formal permission was not needed because its material is already freely licensed.

As a part of the data collection, we did not collect any information about the users aside from their email address which was necessary to verify an account after registration. The data collection was GDPR compliant and we offered to remove any annotations or datapoints belonging to a users should they request that. However, no user made such a request.

As discussed in Section 4, GameQA is a game open to any user in a particular geographic area and does not compensate crowd-workers financially.

Acknowledgements

We thank The Icelandic Web of Science, `mbl.is`, `visir.is`, and the Government Information website for allowing us to include paragraphs, from articles on their websites, in our RuQuAD corpus.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv:1611.09268*.

B. Barla Cambazoglu, Mark Sanderson, Falk Scholer, and Bruce Croft. 2021. A Review of Public Datasets in Question Answering Research. *SIGIR Forum*, 54(2).

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.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. *arXiv:1704.05179*.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy. Association for Computational Linguistics.

Deepak Gupta, Surabhi Kumari, Asif Ekbal, and Pushpak Bhattacharyya. 2018. MMQA: A multi-domain multi-lingual question-answering framework for English and Hindi. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

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*, 7:452–466.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Anna Björk Nikulásdóttir, Þórunn Arnardóttir, Starkaður Barkarson, Jón Guðnason, Þorsteinn Daði Gunnarsson, Anton Karl Ingason, Haukur Páll Jónsson, Hrafn Loftsson, Hulda Óladóttir, Eiríkur Rögnvaldsson, Einar Freyr Sigurðssona, Atli Þór Sigurgeirsson, Vésteinn Snæbjarnarson, Steinþór

Steingrímsson, and Gunnar Thor Örnólfsson. 2022. Help Yourself from the Buffet: National Language Technology Infrastructure Initiative on CLARIN-IS. In *Selected Papers from the CLARIN Annual Conference 2021, Linköping Electronic Conference Proceedings 189*.

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.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. 2019. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6174–6184, Florence, Italy. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting Closer to AI Complete Question Answering: A Set of Prerequisite Real Tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8722–8731.

Kazutoshi Shinoda, Saku Sugawara, and Akiko Aizawa. 2021. Can question generation debias question answering models? a case study on question–context lexical overlap. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 63–72, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vésteinn Snæbjarnarson and Hafsteinn Einarsson. 2022. Natural Questions in Icelandic. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4488–4496, Marseille, France. European Language Resources Association.

Vésteinn Snæbjarnarson, Haukur Barri Símonarson, Pétur Orri Ragnarsson, Svanhvít Lilja Ingólfsdóttir, Haukur Jónsson, Vilhjalmur Thorsteinsson, and Hafsteinn Einarsson. 2022. A Warm Start and a Clean Crawled Corpus - A Recipe for Good Language Models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4356–4366, Marseille, France. European Language Resources Association.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. NewsQA: A Machine Comprehension Dataset. *arXiv:1611.09830*.

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