None of the above: Comparing Scenarios for Answerability Detection in Question Answering Systems

MSc. Thesis

Julio Reyes Montesinos
Master Universitario en Tecnologías del Lenguaje
ETS de Ingeniería Informática
Universidad Nacional de Educación a Distancia

Supervisors
Prof. Dr. D. Álvaro Rodrigo Yuste
Prof. Dr. D. Anselmo Peñas Padilla

September 2022

Abstract

The recent qualitative step in performance of Question Answering (QA) systems has motivated a parallel profusion of new QA datasets intended to benchmark them. However, there have been only limited efforts to study the range of reasoning phenomena in QA, something that would allow for a more thorough evaluation of QA systems. One phenomenon that has not received much attention is answerability. It is important that question answering systems are able to decide whether to not give an answer when the system is unsure. However, most question answering datasets do not include unanswerable questions, and if they do, do not specify the amount of unanswerable questions. To date, there is no QA dataset or guideline available that specifies the optimal amount of unanswerable questions a QA system should see during training. In this work we propose a modification to the popular multiple-choice question answering dataset *RACE* that renders some questions unanswerable, and we study which proportion of unanswerable questions might offer the best results during training and evaluation of a baseline BERT model.

Resumen

El reciente salto cualitativo en el rendimiento de los sistemas de Búsqueda de Respuestas (QA) ha motivado en paralelo la aparición de un gran número de nuevos conjuntos de datos de QA creados para evaluar dichos sistemas. Sin embargo, no hay suficientes estudios acerca del abanico de fenómenos de razonamiento que ocurren en QA, lo cual permitiría una evaluación más completa de los sistemas de QA. Un fenómeno que no ha recibido suficiente atención es la habilidad de no responder. En la práctica, es importante que cuando un sistema de Búsqueda de Respuestas no está seguro, pueda decidir no ofrecer ninguna respuesta. Sin embargo, la mayoría de los conjuntos de datos de QA no incluyen preguntas sin respuesta y, si las incluyen, no especifican en qué proporción. Hasta la fecha, no hay ningún conjunto de datos o guía para su creación disponible que especifique la cantidad óptima de preguntas sin respuesta que un sistema de Búsqueda de Respuestas debería ver durante el entrenamiento. En este trabajo, proponemos una modificación del popular conjunto de datos de opción múltiple *RACE* que convierte algunas preguntas en preguntas sin respuesta, y analizamos qué proporción de preguntas sin respuesta podría ofrecer los mejores resultados durante el entrenamiento y la evaluación de un modelo BERT de base.

Contents

1 Introduction

1.1 Area of study and terminology

Reading Comprehension (RC) is the ability to read text and understand its meaning. Building machines with this ability, or **Machine Reading Comprehension (MRC)**, is one of the most elusive and long-standing challenges in Artificial Intelligence (Norvig, 1986). To define MRC, researchers often invoke the Turing test: a machine reaches human level intelligence if its responses in a dialogue with a human cannot be distinguished over the long haul from those produced by another human (Turing, 1950). While it has been discussed that defining the problem in terms of human competence focuses on a wrong goal, Turing's definition does point out to a workable format to evaluate MRC (Levesque, 2014). Asking questions about a text passage is a common way to evaluate RC in humans, and the same approach has been suggested for testing computers. The task is defined generally in (Burges, 2013): "A machine comprehends a passage of text if, for any question regarding that text that can be answered correctly by a majority of native speakers, that machine can provide a string which those speakers would agree both answers that question, and does not contain information irrelevant to that question."

At the same time, MRC can be placed within the larger, practical problem of **Question Answering (QA)**. QA aims to give a precise answer to a user's question in natural language. Dating back to the 1960s, the first efforts in the field sought to provide natural language interfaces to manually constructed, close-domain knowledge bases. This is called **Knowledge-Based QA**. The other, early paradigm was based on text, and known as **Information Retrieval (IR)-based QA**. The architecture of early IR-based QA systems consisted of different modules such as question analysis, passage retrieval, and answer extraction (Allam and Haggag, 2012). With the rise of neural Natural Language Processing, the typical QA system architecture evolved to a **Retriever-Reader** setup where, given a question, the Retriever finds relevant documents and the Reader performs MRC to extract the answer from them. With earlier neural models such as Recurrent Neural Networks, this meant first encoding the passage and the question separately and then matching them in a combined representation (Qiu et al., 2019). Current approaches are based on Transformers, generic pretrained models which can be applied directly to a variety of MRC tasks, in an end-to-end fashion and with minimal adaptations. These models do away with the aforementioned separation, jointly encoding the passage, the question and the eventual answers. Thanks to Transformer models, recent years have seen a large improvement in QA systems and successful commercial applications are nowadays a ubiquitous reality. General search engines like Google and Bing now integrate Open-domain QA to provide instant answers to queries along traditional search results, while close-domain QA is an essential component of dialogue systems broadly adopted in customer support environments or enterprise FAQs.

Parallel to the improvement in QA systems brought by neural models and Transformers in particular, recent years have seen an explosion in the number

of **MRC datasets** available (Rogers et al., 2021). Besides the need to develop commercial QA systems, a main reason for this abundance is the need to evaluate Transformers. Transformers are neural models pretrained on large corpora with an objective of general language understanding, a vague notion that cannot be measured directly. The reason for their success is their ample scalability, both in the sense that they can leverage knowledge from a vast range of examples almost without supervision, and that they can be applied to numerous NLP tasks. But a lack of interpretability means that it is unclear what Transformers actually learn. Their success at a task can only be explained in terms of the task itself — when they fail, it is not possible to know why they do so. This has fostered efforts to study and define the different skills at play in MRC, pointing at new directions for the development of MRC datasets. The present work can be placed within such efforts.

1.2 Motivation

When specifically proposing the **Multiple-Choice** format to evaluate MRC, (Levesque, 2014) emphasizes that questions and answers should be carefully designed by humans to ensure they require some degree of background knowledge. Such background knowledge would form the basis of a world model, which is a requirement to perform complex inference. Due to pretraining, transformer models already encode much background knowledge. But it is unclear to which extent they use it for inference. For example, while current MRC systems can reliably answer factoid questions — which they do by simple paraphrasing of the context, or perhaps using the large number of factoids the already encode — they do not reliably check that their answers are entailed by the text (Rajpurkar et al., 2018). But for real-world QA systems, the ability to reason about a question's **answerability** given a context is critical: if the answer is not contained in the reference document(s), assuming that a question is answerable leads to a wrong answer. And in the development of MRC datasets, it is critical that they allow diagnosing model performance: did the model provide the right answer because it saw it during pretraining of did it correctly infer it from the text? Did the model provide a wrong answer because it interpreted the text wrongly or was it just obliged to give an answer when it was not confident enough about any of the possibilities?

Until recent, most MRC datasets have assumed that an answer exists for every question. Recent examples have started to address the problem of diagnosing different reasoning skills, among them answerability detection. SQuAD 2.0 (Rajpurkar et al., 2018) proposes unanswerable questions, but in an extractive format: if the answer is not found within the reference text, the model should just provide an empty string as the answer. QuAIL (Rogers et al., 2020) is a recent, multiple-choice MRC dataset with questions annotated by reasoning type, one of which is unanswerable questions. However, it does not particularly target answerability detection, and its proportion of unanswerable questions is simply balanced with respect to other reasoning types. In this work we propose to focus on unanswerable questions within the multiple-choice format and study the

effect of different proportions of unanswerable questions in a dataset.

1.3 Research questions

To state the research problem, we break down the motivation into the following research questions:

1. What is the optimal scenario to identify unanswerable questions?
2. Does BERT need to see a higher proportion of unanswerable questions in training to be able to reliably identify them in test?
3. How does a trained BERT model respond when it is tested on different proportions of unanswerable questions?

1.4 Document outline

In this work we propose a method that modifies the popular multiple-choice MRC dataset RACE (Lai et al., 2017) to include unanswerable questions, and study the performance of a baseline BERT model in different training and evaluation scenarios with unanswerable questions. We accomplish this by (1)creating a range of copies of RACE with different proportions of unanswerable questions, (2)fine-tuning a BERT model on each of the modified copies, and (3)evaluating each of the fine-tuned models on each of the modified copies of RACE.

The rest of this document is structured as follows: In chapter 2, we survey the state-of-the-art of MRC datasets, with special attention to strategies to include unanswerable questions, and describe the use of Transformers in QA. Chapter 3 introduces the dataset and presents the modification method in detail. Chapter 4 is dedicated to studying the evaluation results. We draw conclusions from that study in Chapter 5.

2 State of the art

2.1 Historical approximation to Question Answering and Machine Reading Comprehension

Question answering (QA) has been a fundamental research area in Natural Language Processing, inherent to natural language interfaces for information systems. At present, QA systems are deployed in contexts such as virtual assistants, search engines, or database interfaces. As a general notion, QA attempts to automatically provide an exact answer to a given natural language question. This notion can be formalized as several concrete tasks depending on the format of the question, the format of the answer and the resources where the answer should be found.

Starting from the 1960s, there have been two major paradigms of question answering (QA): Knowledge-Based QA and Information Retrieval QA. The best known early systems, Baseball (Green Jr et al., 1961) and LUNAR (Woods, 1972), are two examples of knowledge-based QA operating over the restricted domains of the US baseball league and the Moon's geological facts, respectively. Knowledge-based QA focuses on mapping natural language questions to a formal representation that can be used to query a database in a certain, closed domain.

Conversely, Information Retrieval (IR) works over non-restricted domains. IR-based QA can be seen as a two-stage task where, given a query, an information retrieval algorithm selects relevant documents or passages from a collection, and a reading comprehension algorithm selects an answer from these passages in the form of a span of text. These two steps can be generalized as answer generation and answer *validation*. An early example of this two-step architecture is (Simmons et al., 1964). Early IR algorithms of this type were hindered by the vocabulary mismatch problem (Furnas et al., 1987; Jones, 1972), since they expected an overlap of words between the question and the passage. This limitation was partially alleviated by dense word representations that could handle synonymy, starting with Latent Semantic Indexing (Deerwester et al., 1990) up until the current contextual word embeddings (Devlin et al., 2018). A more flexible semantic overlap between question and passage is very beneficial for factoid questions, which require little reasoning and can be solved by semantic entailment in semantic relations are encoded more generally.

In contrast with factoid questions that can be solved by extractive IR, complex QA requires integrating information and reasoning about events, entities, and their relations across multiple sentences in the document, which can be understood as general reading comprehension. At present, the focus of the QA community has moved away from information retrieval (i.e. the answer generation step), and onto reading comprehension (i.e. question validation) (Rodrigo and Penas, 2017). This shift can be illustrated by the introduction of the Question Answering for Machine Reading Evaluation (QA4MRE) campaign (Peñas et al., 2012) at the Conference and Labs of the Evaluation Forum (CLEF) in 2011. The QA4MRE campaign was based on Multiple Choice Reading Comprehension

datasets, which simulate a scenario where candidate answers have already been generated, so the focus is on validating these answers.

The name *Question Answering for Machine Reading Evaluation* reflects the current importance of QA as the format (Gardner et al., 2019) of choice to evaluate the reading comprehension capability of current neural models aimed at general language understanding. Transformer models' dominance and the necessity of tasks to evaluate them justify the focus on answer validation, meaning that at present the tasks of **QA and Machine Reading Comprehension (MRC) largely overlap** (Chen, 2018). In the literature, similar datasets and systems are described as QA or MRC indistinctively. And while it can also be discussed (Zhang et al., 2020d) that MRC also encompasses other tasks beyond QA, such as textual entailment, this work focuses on MRC understood as the answer validation component of QA.

2.2 Task definition

Machine Reading Comprehension requires inferring the answer to a given question from a given context alone, even when the syntactic match between question and passage is poor. The task can be formulated as a supervised learning problem where given a collection of training examples $\{(p_i, q_i, a_i)\}_{i=1}^{n}$, the goal is to learn a predictor $f : (p, q) \longrightarrow a$ which takes a passage of text p and a corresponding question q as inputs and gives the answer a as output. This formulation (Chen, 2018) then adopts different forms: the passage p can be a short paragraph or a document with several paragraphs, the notion of question q can be expanded to a cloze-style (fill-in-the-gap) task, and the task can involve extracting a span of text from the passage, choosing an answer among multiple options, or even generating a free-form answer.

2.3 Task types and key datasets

Due to their having been generally adopted for the evaluation of systems capable of general reading comprehension and the steep evolution of these, recent years have seen an explosion in the number of Machine Reading Comprehension datasets available. As of September 2022, the popular huggingface repository[1] lists 266 datasets in the *question answering* category. An exhaustive analysis of this landscape is beyond the scope of this work, but in this section, following (Chen, 2018)'s taxonomy, we will outline the types of QA/MRC tasks and describe a paradigmatic English dataset of each type. More complete surveys can be found in (S. Liu et al., 2019; Rogers et al., 2021; Sugawara et al., 2018; Zhang et al., 2019; Zhang et al., 2020d).

[1] `https://huggingface.co/datasets?task_categories=task_categories:question-answering`

2.3.1 Cloze-style

In contrast to formal questions, cloze-style queries consist of a sentence with one or more gaps, so the answer is a word or short span of word, often representing an entity or fact mentioned in the passage. The first large-scale MRC dataset, the CNN/Daily Mail dataset (Hermann et al., 2015), follows this format. It collects over 93k articles from the CNN and 220k articles from the Daily Mail. In the original publications, each article was summarized by a series of bullet points. From these, the authors automatically generate cloze-style queries by replacing one named entity by a placeholder. This process limits the kind of answer — the answers are always entities to be extracted from the article. (Chen et al., 2016) show that in the CNN/Daily Mail dataset, "the required reasoning and inference level …is quite simple" and that a relatively simple algorithm can get almost close to the upper-bound. The CNN/Daily Mail dataset does not deliberately include multi-sentence reasoning or unanswerable questions.

The automatic question-answer generation process is improved on *Who did what* (Onishi et al., 2016), where answer options are person named entities extracted from a related article about the same event. This reduces the syntactic similarity between the question and the passage, increasing the need for deeper semantic analysis. Still, the automatic question generation process introduces a significant amount of noise in the dataset and limits the ceiling performance by domain experts (Lai et al., 2017). One more relevant cloze-style dataset is CLOTH (Xie et al., 2017), where human-crafted questions have been collected from real reading comprehension tests for students. This method is discussed in the next point.

2.3.2 Multiple-choice QA

In the Multiple-choice QA task, each question is paired with k hypothesized answers (usually $k = 4$) where only one is correct. Research (Sugawara et al., 2018) shows that multiple-choice questions tend to require a broader range of reasoning skills than answer extraction questions, which usually can only target factoids. On the other hand, the multiple-choice format can be seen as artificial regarding real-world scenarios, since it transforms the problem from giving the correct answer to choosing the most plausible candidate. Still, the multiple-choice format is considered an acceptable compromise between extractive and free-form answers, given that it is still too difficult to evaluate the latter (Rogers et al., 2020). The canonical large-scale dataset of this type is RACE (Lai et al., 2017), which collects English as a second language exams at two levels of difficulty. Since the RACE dataset is the object of this work, it is described more in-depth in Chapter 3.

The idea of using real-world human tasks to evaluate machine reading comprehension systems was introduced in (Hirschman et al., 1999), which collects children's reading comprehension tests with five "wh" (what, where, when, why, and who) questions. (Levesque, 2014) suggest specifically the multiple-choice format to evaluate MRC, emphasizing that, to ensure they demand complex rea-

soning, questions and answers should be carefully designed by humans. The Winograd schema (Levesque et al., 2012) proposes single sentences as reference, paired with binary questions. Another precursor is MCTest (Richardson et al., 2013), still at a small scale (500 stories and 2000 questions). (Rodrigo et al., 2015) collects more challenging questions from University entrance examinations, that require a higher degree of textual inference, but as an evaluation task (24 passages and 115 questions) it is not a sufficient training source for neural models.

2.3.3 Extractive QA

The extractive QA task is equivalent to the aforementioned reading comprehension step in IR-based QA: the answer a must be a single span in the given passage string p, and can be represented by positions a_start and a_end in p. The canonical extractive QA dataset is the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016). SQuAD was the first large-scale reading comprehension dataset composed of natural questions — written by humans. It contains 107k crowd-sourced question-answer pairs over 536 Wikipedia articles. SQuAD has become the central benchmark in the field, with most models reporting evaluation results on it.

The obvious limitation of extractive QA datasets is that they can only target information explicitly mentioned in the text, and therefore can often get solved with shallow lexical matching. In SQuAD, most questions have answer spans embedded in sentences that are syntactically similar to the question, and this syntactic similarity becomes a reliable shortcut that the model learns during training. Another practical limitation of SQuAD is the lack of unanswerable questions, caused by the recollection directive of formulating questions whose answer can be found in the related passage. This problem is addressed in a subsequent version, SQuAD 2.0 (Rajpurkar et al., 2018).

A different construction process is on TriviaQA(Joshi et al., 2017), based on question-answer pairs collected from trivia websites. Evidence passages are gathered from Wikipedia after the collection of questions, resulting in a notably more challenging dataset than SQuAD due to the considerable syntactic and lexical variability between questions and the corresponding evidence.

2.3.4 Freeform QA

The last category of QA tasks does not pre-specify any answer options, allowing instead answers that consist of an arbitrary sequence of words. To answer the questions, the machine needs to reason across multiple pieces of the text and summarize the evidence (Chen, 2018). The milestone dataset in this category is MS MARCO (Nguyen et al., 2016), that contains 1M questions sampled from real, anonymized user queries to Microsoft's Bing search engine paired with passages extracted from real web documents retrieved by Bing. A similar example is Natural Questions (Kwiatkowski et al., 2019), with queries from the Google search engine. Another notable example is NarrativeQA (Kočiskỳ

et al., 2018), which consists of summarized narratives from books and movie scripts extracted from Wikipedia and question-answer pairs about them written by human workers. Particular about NarrativeQA is that answering the questions requires understanding the underlying narrative rather than relying on shallow pattern matching or salience.

Although the free-form task is the most similar to real application scenarios, it also the most difficult to evaluate because it requires special metrics, such as BLEU or ROUGE, to assess the natural language generation component.

2.4 Problems of existing MRC datasets

2.4.1 Natural questions

One of the challenges of constructing large-scale datasets is obtaining the questions. Most datasets in the previous section generate the questions through automatic processes or crowdsourcing, in both cases based on the evidence. This hampers the applicability of experiments to real-world scenarios, where users information needs are spontaneous and unconstrained. Since they do not include restricted answers, datasets based on naturally occurring questions such as the aforementioned (Kočiskỳ et al., 2018; Kwiatkowski et al., 2019; Nguyen et al., 2016) are more difficult to benchmark. An initiative around this problem is BoolQ (Clark et al., 2019), which follows the data collection method described in (Kwiatkowski et al., 2019) but only includes yes/no questions. Besides facilitating evaluation, yes/no questions have the benefit of often requiring a wide range of inferential abilities to solve.

2.4.2 Complex reasoning

As MRC systems reach human performance on the most popular MRC datasets, different strategies has been followed to create more difficult datasets. For example, the ARC dataset (Clark et al., 2018) discards questions if they are too easy for a word co-occurrence algorithm, and ComQA (Abujabal et al., 2018) discard questions whose answer could be found by existing search engine technology. Other datasets focus on specific types of reasoning, such as counting or sorting (Dua et al., 2019) or coreferential (Dasigi et al., 2019).

A principled approach to categorizing reasoning difficulty was taken in (Chen et al., 2016; Lai et al., 2017), which establish five levels of reasoning difficulty (in increasing order): word matching, paraphrasing, single-sentence reasoning, multiple-sentence reasoning and insufficient/ambiguous. According to (Lai et al., 2017), many questions in popular MRC datasets like CNN, SQuAD, and RACE are simple factoid questions, or anyway can be solved by simple word matching or paraphrasing (Lai et al., 2017). Single-sentence reasoning is a lot easier for models than multi-sentence reasoning, as shown in (Richardson et al., 2013). Integrating the information contained in multiple sentences is also much more difficult for humans (Berninger et al., 2011). A dataset that focuses on multi-sentence reasoning is MultiRC (Khashabi et al., 2018), and this concept is

extended to long documents in NarrativeQA (Kočiskỳ et al., 2018), and multiple documents (Yang et al., 2018). A comprehensive approach to several reasoning phenomena is QuAIL (Rogers et al., 2020), a multiple-choice QA dataset where questions are annotated by type of reasoning skill. QA system might face different sources of uncertainty; a taxonomy of complex reasoning skills needed is detailed in (Rogers et al., 2021):

- Inference: refers to moving from a set of propositions to an accepted answer. Several types of inference are categorized in philosophy, attending to the kind of support (analogy, best explanation, defeasible reasoning), the strength or the direction (inductive, abductive).
- Retrieval: while the extent of retrieval in MRC datasets may only refer to the problem of locating information within a passage, another dimension of retrieval is **answerability**, i.e. deciding whether the necessary information can be found within the passage. We discuss this phenomenon separately below.
- Input interpretation and manipulation: an MRC system needs the linguistic skills to interpret coreference and ellipsis, as well as the numeric skills to perform basic counting and ordering.
- Background knowledge: refers to a system's access to the multitude of facts necessary to create a world model upon which to base its common reasoning. This phenomenon is discussed in the following subsection.
- Multi-step: a question may combine any of the above dimensions, that need to be properly chained.

2.4.3 Background knowledge

An identifiable source of uncertainty when evaluating QA systems is background knowledge. For example, if a question refers to something that happens *several times a month* and the supporting passage mentions that an event happens *every Saturday*, the reasoning involved to answer the question would be supported by the external knowledge that there are several Saturdays in each month. In (Rogers et al., 2020), questions involving causality, entity properties, belief states, subsequent state, or event duration are characterized as requiring world knowledge to be answered.

QA systems need to incorporate world knowledge to answer these questions, but at the same time, system evaluation needs to account for the possibility that questions may be (partially) answered by the system's world knowledge. In the past, QA systems have tried to incorporate world knowledge using resources such as *Wordnet* or *FrameNets*. By contrast, current QA systems based on large language models already encode much background information due to pretraining, and in fact be queried for factoids directly (Roberts et al., 2020). Their evaluation poses the challenge of discriminating whether the system is performing

inference over the given passage or utilizing pre-encoded background knowledge to answer questions. Another reasons why QuAIL includes reasoning type annotations is to examine this question.

2.4.4 Unanswerable questions

MRC datasets such as (Hermann et al., 2015; Lai et al., 2017; Rajpurkar et al., 2016) are constructed with the assumption that a correct answer for every question exist within the given context. However, this assumption does not hold in real-world QA applications. For example, in web search there can be multiple possible sources of information (typically web snippets) that may or may not contain the correct answer. Since the information contained in a given passage is limited, some questions inevitably will have no answer within the context of that passage. In such scenario, a robust MRC system should refrain from answering rather than make an unreliable guess — a sign of language understanding ability. But to be able to consider the possibility that some questions have no answer, models need to be trained on unanswerable examples.

Adding onto the previous version of SQuAD, SQuAD 2.0 (Rajpurkar et al., 2018) includes more than 50k unanswerable questions written by crowd-workers. The premise was to add relevant questions with plausible (yet incorrect) answers within the given passage, but which were unanswerable based on the passage alone. Their analysis shows that systems' performance is overestimated in the presence of unanswerable questions.

Adversarial training Earlier studies have used automatic adversarial methods to probe model robustness with similar conclusions. (Jia and Liang, 2017) show how model performance on SQuAD degrades by more than half when tested over examples adversarially modified with their *AddSent* algorithm, which appends a sentence that resembles the question to the reference passage. However, the data generated this way is similar to the original, resulting in a less diverse test set. (Wang and Bansal, 2018) propose an improved version of the algorithm, *AddSentDiverse*, and an improved training regime including adversarial data augmentation. (Gan and Ng, 2019) propose adversarial question paraphrasing to test models' reliance on string matching, and also apply the method to the creation of training data, improving on models' robustness. (Ziqing Yang et al., 2019) experiment over both SQuAD and RACE, but instead of corrupting the datasets they apply adversarial perturbations at the level of word embeddings during training.

In contrast with SQuAD 2.0, these adversarial methods have the advantage of needing less human labour. However, they do not necessarily produce unanswerable questions. An exception is (Zhu et al., 2019), but the question variations produced are too lexically similar to the original ones and therefore do not clarify whether the model fully understands them or relies on superficial cues.

Answer removal While the above adversarial methods work by producing modified questions for extractive MRC, other dataset formats allow simpler methods. (Pradel et al., 2020) shows an example of unanswerable question generation in Knowledge-Based QA. They modify the Spider KB question answering dataset by deliberately removing some information from the underlying relational databases. The present work follows a similar approach over the multiple-choice MRC format.

Modelling question answerability MRC with unanswerable questions takes into consideration the possibility that some questions cannot be answered based on the given context. Here there are two subtasks to consider: giving the accurate answer for answerable questions, and effectively distinguishing answerable from unanswerable questions. This last subtask requires an extra answer verification mechanism. This verification step can be as simple as establishing a threshold for prediction confidence in general purpose CLMs (Devlin et al., 2018; Zhang et al., 2020a) and giving a null answer when that threshold is not reached. Another option is to add a "no-answer" score to the score vector of the answer span, as in (Clark and Gardner, 2017). A dedicated verifier to determine answerability can be based on a combined representation of the passage and the answer, as in (Sun et al., 2018) and (Tan et al., 2018). Dedicated verifiers can also be additional layers to be trained jointly in a multitask-learning setup, as in (Liu et al., 2017). In addition to this, (Hu et al., 2019) takes one more step to verify whether the predicted answer is entailed by the reference passage. This approach is also seen in (Back et al., 2019), who inspect whether the answer meets all the conditions extracted from the question by comparing the embeddings of both, which allows explaining why a question is classified as unanswerable. In (Zhang et al., 2020b), such restrictions are modelled as syntactic constraints. Finally, a multitask-style approach inspired by human reading comprehension strategies can be seen in (Zhang et al., 2020c), where a "sketchy" reading of the relationship between passage and question is done in parallel to an intensive reading that verifies the answer.

2.5 An overview of current RC systems: Transformer models

Reading comprehension datasets and models have evolved in parallel: understanding the performance of existing models helps identify the limitations of existing datasets (Chen, 2018), and more challenging datasets call for more advanced models.

Since 2018, a new generation of neural models has brought substantial advancement to most NLP tasks. The transformer is a novel neural network architecture for sequence modelling. It follows the **encoder-decoder**, end-to-end framework seen in previous state-of-the-art recurrent neural network models like LSTMs (Seo et al., 2016) or gated networks (Wang et al., 2017), but crucially dispenses with recurrence or convolutions, instead being based entirely on self-attention (Vaswani et al., 2017). Most importantly, this allows for **paral-**

lelization and thus faster training times. Fundamental for reading comprehension, transformers are also better than LSTMs at modelling long-term dependencies (Dai et al., 2019) and word sense disambiguation (Tang et al., 2018). Transformer models are first trained on very large, unlabelled text corpora with a language modelling objective, in a process known as **pretraining** (Dai and Le, 2015; Erhan et al., 2010). This self-supervised (Vaswani et al., 2017) step ensures that transformers become universal approximators (Yun et al., 2019) that can later be adapted to a variety of domains and downstream tasks by **fine-tuning** (Howard and Ruder, 2018) on smaller datasets, without substantial task-specific changes to the model architecture. The potential to leverage linguistic information from the vast amount of unlabelled text data available today has made transfer learning the current most common approach to many language understanding tasks (Tenney et al., 2019).

Based on this shared high-level idea, different pretraining objectives have been explored, with autoregressive and autoencoding language modelling being the most successful ones. Autoregressive language modelling aims to estimate the probability of a text corpus by factorizing the likelihood of a text sequence $\mathbf{x} = (x_1, \ldots, x_T)$ into a forward product $p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t|\mathbf{x}_{<t})$, thus only learning uni-directional representations. By contrast, autoencoding-based pretraining does not explicitly perform density estimation, but instead seeks to reconstruct the original data from corrupted input (Vincent et al., 2008).

The model we use in this work is BERT (Devlin et al., 2018), the most notable example of autoencoding-based pretraining and currently most popular baseline approach. BERT's main pretraining objective, the **masked language model**, consists in randomly replacing 15% of the tokens from the input by a special symbol `[MASK]` (or sometimes another random word) and trying to predict the original tokens. This effectively fuses left and right contexts. The capability of modelling deep bidirectional contexts is crucial for QA tasks. An additional training objective, **next sentence prediction**, emphasizes the relationships between two consecutive sentences, and thus makes BERT even more suitable for QA tasks.

More performant, autoregressive models such as XLNet (Dai et al., 2019) have appeared after BERT. However, BERT is still much less computationally expensive because its training objective dispenses with density estimation (instead, BERT makes the unfavourable assumption of independence between masked tokens), and therefore remains the de facto baseline approach to QA tasks.

3 Methodology

3.1 The RACE Dataset

The original RACE dataset (Lai et al., 2017) is a canonical benchmark in Multiple-Choice Reading Comprehension (Devlin et al., 2018; Lan et al., 2019; Y. Liu et al., 2019; Zhilin Yang et al., 2019). RACE is a collection of real English as a Second Language exams for 12- to 18-year-old students in China. The exams are intentionally designed by human experts (English instructors) to evaluate human language understanding and reasoning, which makes RACE an adequate tool to examine Machine Reading Comprehension systems too. The dataset is also large enough to allow the training of current data-driven Machine Reading Comprehension systems.

Passage:

In my second year of high school, the class was scheduled to run the mile. when the coach yelled, "Ready. Set. Go!", I rushed out like an airplane, faster than anyone else for the first 20 feet. I made up my mind to finish first. As we came around the first of four laps, there were students all over the track. By the end of the second lap, many of the students had already stopped. They had given up and were on the ground breathing heavily. As I started the third lap, only a few of my classmates were on the track. By the time I hit the fourth lap, I was alone. Then it hit me that nobody had given up. Instead, everyone had already finished. As I ran that last lap, I cried. And 12 minutes, 42 seconds after starting, I crossed the finishing line. I fell to the ground. I was very upset.
Suddenly my coach ran up to me and picked me up, yelling, "You did it. Mark! You finished, son. You finished" He looked at me straight in the eyes, waving a piece of paper in his hand. It was my goal for the day which I had forgotten. I had given it to him before class. He read it aloud to everyone. It simply said, "I, Mark Brown, will finish the mile run tomorrow, come what may." My heart lifted. My tears went away, and I had a smile on my face as if I had eaten a banana. My classmates clapped. It was then I realized winning isn't always finishing first. Sometimes winning is just finishing.

Questions (correct answer in **bold**):

1) It took Mark _ to run the mile.
A. about 13 minutes
B. more than 13 minutes
C. only 12 minutes
D. less than 12 minutes

1) Why did Mark cry when he ran the last lap?
A. Because he was quite happy.
B. Because he was too upset.
C. Because he got a pain in his heart.
D. Because he was hungry.

Figure 1: Original sample passage and corresponding questions from RACE.

The collected exams consist of a supporting passage accompanied by a variable number of questions about it. Each of these questions is in turn paired with 4 candidate answers, of which only one is correct. A sample passage and two corresponding questions from RACE-M can be seen in Figure 1.

The exams originate from either middle- (12 to 15 years old) or high-school (15 to 18) examinations, thus allowing the dataset to be separated in two levels of difficulty (which the authors denominate RACE-M and RACE-H, respectively). There is a wide gap in difficulty; passages, questions and candidate answers in RACE-H are 52% longer on average, and contain a much wider vocabulary (125120 tokens in RACE-H vs. 32811 in RACE-M). The authors claim that, since both its questions and candidate answers are human generated, RACE is more challenging than comparable-scale Reading Comprehension datasets. To support this claim, they annotate a sample of questions with the type of reasoning phenomena involved: statistics show that 33% of the questions in RACE involve single-sentence reasoning and 26% multi-sentence reasoning, while a combined 37% can be solved with word matching or paraphrasing – this last figure is 74% for SquAD. However, in contrast with QuAIL, the RACE dataset is not annotated with reasoning type beyond this analysed sample.

RACE contains a total of 27933 text passages with 97687 questions. The authors provide predefined train, validation and test splits. The tables below detail the numbers of passages (table 1) and questions (table 2) per difficulty level and split:

Table 1: Number of passages per difficulty level and split in RACE.

division	train	validation	test	all
RACE-H	18728	1021	1045	20794
RACE-M	6409	368	362	7139
all	25137	1389	1407	27933

Table 2: Total number of questions per difficulty level and split in RACE.

division	train	validation	test	all
RACE-H	62445	3451	3498	69394
RACE-M	25421	1436	1436	28293
all	87866	4887	4934	97687

3.2 Rendering questions unanswerable

Our definition of unanswerable question is akin to the one seen in QuAIL (Rogers et al., 2020), where a question is annotated as unanswerable when the supporting passage does not provide sufficient information, and world knowledge does not make one of the answers more likely. By contrast, all questions in the original

RACE dataset are answerable in principle. To render a question unanswerable, we simply replace the correct answer option with a sentence that implies that no answer exists among the given options, i.e. *None of the answers are correct*. Eliminating the correct answer turns a question unanswerable regardless of the type of reasoning involved: it works both in cases where the answer is contained in the supporting passage literally or can be inferred by reasoning over it, and questions that require eventual world knowledge to be answered. The remaining three answer options will be plausible but incorrect, thus the only correct answer is choosing that *None of the answers are correct*.

To prevent model overfitting (i.e. that systems learn to identify *None of the answers are correct*. as the correct answer to any question) and again following QuAIL, we also introduce the "unanswerable" option in questions that should remain answerable. In these cases, we replace one of the incorrect answer options chosen at random, therefore keeping the correct answer choice available, but at the same time introducing a different kind of distractor, one that indicates that the question may be unanswerable given its particular context and the other answer choices. Figure 2 shows how we modify the two questions from the previous Figure 1.

A question modified to be unanswerable:

1) It took Mark _ to run the mile.
A. None of the answers are correct.
B. more than 13 minutes
C. only 12 minutes
D. less than 12 minutes

A question modified to remain answerable:

1) Why did Mark cry when he ran the last lap?
A. Because he was quite happy.
B. Because he was too upset.
C. Because he got a pain in his heart.
D. None of the answers are correct.

Figure 2: Modified sample questions from RACE.

3.3 Generating a modified version of RACE

For convenience, we access the RACE dataset through the huggingface's Datasets (Lhoest et al., 2021) library, where instances are already separated by question (in the version provided by the author, questions are grouped together with their common supporting passage). In this structure, each example consists of a question, supporting passage, four candidate answers, and label of the correct answer.

We create a series of altered versions of the original dataset in order to simulate scenarios with a different, measurable occurrence of unanswerable questions. For every version, we apply the modification procedure described in the

previous section to all questions in the dataset. A parameter C governs the rate of unanswerable questions in a version, and thus the probability of eliminating (replacing) the correct answer choice. We divide the dataset by split and difficulty level, and apply the parameter to each group separately, choosing $C \times N$ examples at random, where N is the number of questions in a particular difficulty level and split. On these chosen instances, we replace the correct answer by *None of the answers are correct.*. On the rest of the instances, we preserve the correct answer and replace an incorrect candidate at random.

For test splits, the process is repeated 5 times, creating 5 test splits per dataset with differently altered instances.

The value of the parameter C is in the range $[0 - 1]$, where 0 indicates that the replaced option will always be an incorrect one and therefore all questions remain answerable, and 1 indicates that for all questions the correct option will be replaced, producing a scenario where all questions become unanswerable. Intuitively, these extreme scenarios are senseless, and we expect the middle values of C to produce the interesting results. Still, the aim of the experiment is to compare all possible scenarios. We give C the whole range of values $[0 - 1]$ in steps of 0.1, producing 11 modified copies of RACE with proportions of 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, and 100% of unanswerable questions.

3.4 Data pre-processing

A large portion of questions in RACE are not proper questions but cloze tasks, where a gap in a sentence must be filled with a word or short span of words. Candidate answers to cloze tasks usually do not constitute fully formed sentences. We identify cloze tasks by the character "_", used to signal the gap to be filled. By contrast, proper questions usually contain the character "?". We count the questions containing "_" and/or "?" (see Table 3) and manually examine questions that contain both or none and their corresponding answers, deciding to treat all questions containing "_" as cloze tasks and questions not cntaining that character as proper questions.

Table 3: Number of questions in RACE, per difficulty level and split, containing the characters "_" and/or "?".

level	split	contains "?"	contains "_"	contains "?" and "_"	neither "?" nor "_"
RACE-H	train	29438	31340	557	1110
	validation	1610	1737	33	71
	test	1588	1815	33	62
RACE-M	train	10965	13629	549	278
	validation	620	771	34	11
	test	617	774	18	27

BERT needs to be fed sequences of sentences separated by a special token, `[CLS]`. Thus, to feed the model we need to transform the dataset's instances from a set `{passage, question, 4 options, answer}` to a sequence. The generation of this sequence depends on the type of questions. For proper questions, the resulting sequence has three items, of the form `[passage, question, option]`. By contrast, for cloze tasks we substitute the answer within the question, obtaining a sequence with two items of the form `[passage, question+option]`. For the answer option that has been replaced, "proper question" instances have the form `[passage, question, None of the answers are correct.]`, while cloze tasks have the form `[passage, None of the answers are correct.]`

3.5 Training and evaluation

On each of the 11 datasets we fine-tune a pre-trained BERT model with the same hyperparameters. We fine-tune on the train splits of both RACE-M and RACE-H.

Each of these 11 models is evaluated on the 11 datasets. For each dataset, the model is evaluated on 5 different test splits. The battery of metrics is calculated separately for each of these 5 test splits, and then averaged by dataset.

3.6 Experimental setup

We use the English BERT-base[2] from huggingface Transformers (Wolf et al., 2019) in a Google Colab[3] instance with 8 TPUv2 cores. Furthermore, we make use of the PyTorch framework (Paszke et al., 2019) and huggingface's Datasets (Lhoest et al., 2021) library. Additionally, training was aided by huggingface's Accelerate[4] library while in evaluation we used PyTorch/XLA[5].

[2] `https://huggingface.co/bert-base-uncased`
[3] `https://colab.research.google.com`
[4] `https://huggingface.co/docs/accelerate/index`
[5] `https://github.com/pytorch/xla/`

4 Experimental results

In this chapter, we apply a battery of metrics to the evaluation results obtained in the previous chapter and discuss the results. For every metric, we construct an 11×11 matrix that relates the 11 models (each trained with a differently modified version of the *RACE* dataset) with the 11 test sets. The intention is to compare the evaluation results of every model over every test set in a single overview, all at once.

We present these matrices of results as heatmaps (colour-shaded tables) where columns represent the 11 trained models ordered by the percentage of unanswerable questions on their training set, and rows represent the 11 test sets — also ordered by the percentage of unanswerable questions in them. In this setup, a cell contains the value of a metric calculated on the evaluation results of a particular model over a particular test set. For instance (see Figure 3), on the heatmap for recall, cell $(7, 4)$ contains the value of recall obtained after evaluating the model we trained on a modified version of *RACE* with 70% of unanswerable questions over a modified version of *RACE* with 40% of unanswerable questions.

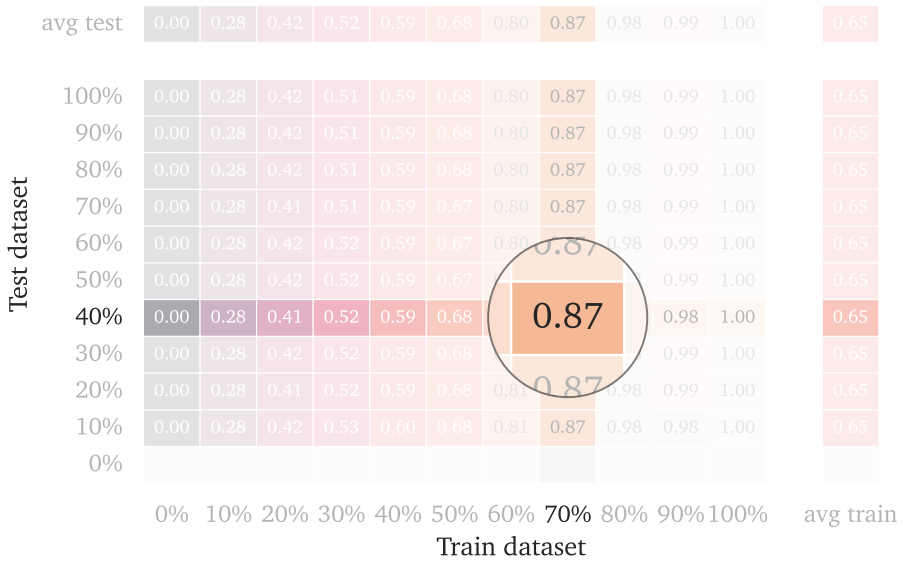

Figure 3: Detail of a cell in the matrix of Recall values: the zoomed-in cell displays the Recall value obtained by testing the model trained on 70% or unanswerable questions on a test set with 40% of unanswerable questions.

All the discussed produce values in the range $[0, 1]$. We display metric values as float point numbers with a precision of 2 decimals — we sacrifice the convenience of converting to percentage to avoid confusion with axes values. Cells are shaded according to the value they contain, with values close to 0 in black

and values close to 1 in lighter colours. We use two different colour schemes:

(a) When introducing metrics based on **accuracy**, i.e. whether the predicted answer is correct, we use the *mako* colourmap included in the visualization package Seaborn[6], where mid values map to bright blues and high values map to lighter green.

(b) When presenting metrics based on **answerability**, i.e. whether the question is predicted to be answerable or unanswerable, we use Seaborn's *rocket* colourmap, where mid values map to bright reds and high values map to lighter orange.

To allow visual comparison of ranges and values among different metrics, mapping remains constant among heatmaps, i.e. the colourmap always maps to the range $[0 - 1]$, and a particular value, such as 0.4, will have the same colour in every table.

In tables presenting answerability-based metrics, we shall see that some cells are light grey. This indicates that their value is undetermined. These are caused by zero division. For example, if we look at the *recall* of unanswerable questions (Figure 6), the bottom row is greyed out, since it represents a test set with zero unanswerable questions, and therefore calculating recall here involves zero division.

At the upper and right ends of every heatmap we show column and row averages, respectively. Column averages allow an overview of the behaviour of a certain model across multiple testing scenarios, while row averages sum up the difficulty of a test set for different models.

We have evaluated the models on (modified versions of) *RACE high* and *RACE middle* separately (models were trained on both), but the results we discuss on this section are aggregated for the entire dataset. The results over (a version of) *RACE middle* are always better than the results over (the corresponding version of) *RACE high*. However, we will dismiss these differences in performance because our interest lies in comparing training strategies, which means looking at metric values *within* a certain matrix of models and test sets. For any given metric, the *patterns* that emerge in the matrices for both levels of difficulty are always similar. Therefore, in this section we will be showing aggregated results for *RACE high* and *RACE middle*. For a breakdown by difficulty level, please refer to the Appendix.

[6] `https://seaborn.pydata.org/tutorial/color_palettes.html`

4.1 General accuracy

Multiple-choice question answering systems are usually evaluated according to their general accuracy, i.e. the proportion of questions in a test set for which the predicted answer is equal to the correct answer in relation to the number of questions in the set (N):

$$\text{General acc.} = \frac{\text{correctly answered}}{N}$$

We show the values of this metric for each combination of model and test set on Figure 4. The bottom left cell displays the accuracy of a BERT model that has seen 0% unanswerable questions, neither during training nor during evaluation, and therefore is the baseline value. However, remember that in any given dataset, all instances have been modified — unanswerable questions are those where the correct answer is *None of the answers are correct*.

Figure 4: Model accuracy on modified *RACE* test sets

The general observation on this table comes from looking at the diagonal starting from the bottom-left: a model's accuracy is better when it is tested on a dataset with an amount of unanswerable questions similar to the dataset on which it was trained.

For models trained on datasets with a high amount (80-100%) of unanswerable questions, the accuracy on any particular test set almost matches the amount of unanswerable questions in that set. This suggests that these models

have learnt to identify the "unanswerable" option as the correct answer, and they fail to discern the small percentage of truly answerable questions.

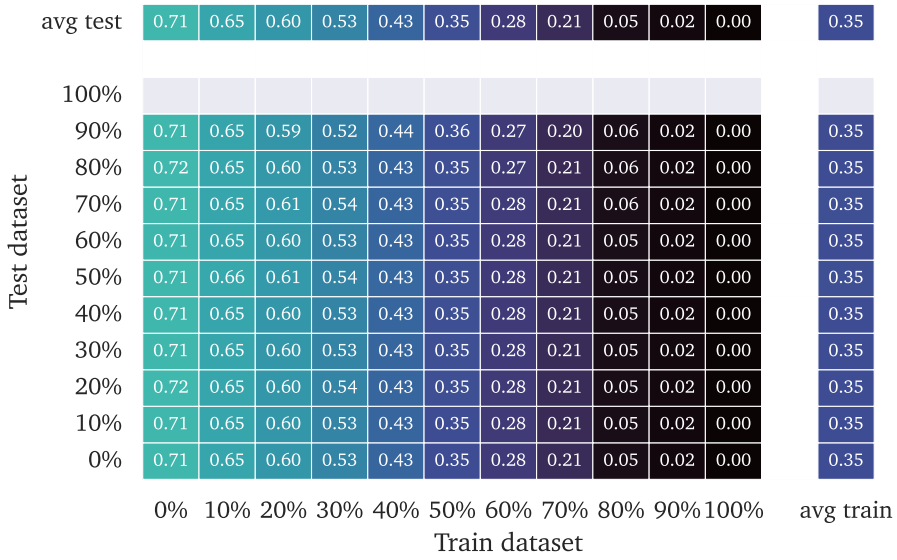

Figure 5: Model accuracy on modified *RACE* test sets when taking only answerable questions into account.

We have intentionally created imbalanced datasets in terms of answerability, and that makes it difficult to compare accuracy values across them. To further break down these results, we will look separately at the accuracy in each group of questions, answerable and unanswerable. Figure 5 shows the general accuracy when only taking answerable questions into account.

Here, we observe that model accuracy remains relatively constant across test sets (i.e. by column), but declines rapidly across models as the percentage of unanswerable questions seen in training rises (i.e. towards the right side of the table). The reason for this is that predictions are independent of each other, thus when only looking at answerable questions, the amount of unanswerable questions in a test set does not matter: what we are looking at here is each model's ability to correctly answer answerable questions. And this ability is severely impacted by the presence of unanswerable questions in training: models trained on over 80% of unanswerable questions are almost completely unable to give proper answers.

While for each model we see a slight change in accuracy on answerable questions towards the upper end of the table, this is likely an artifact due to the smaller number of total answerable questions on those test sets.

Figure 6: Accuracy on unanswerable questions only, or recall, on modified *RACE* test sets.

If instead we look at the accuracy on unanswerable questions (Figure 6), we see the reverse pattern: models trained on a high proportion of unanswerable question can reliably detect them. The models that saw over 80% of unanswerable questions in training can almost always detect them, but as we saw earlier (Figure 5) this is at the expense of the ability to deal with answerable questions. On the other hand, on the left-most column we see that the model that saw 0% of unanswerable questions in training never detects them, suggesting the model's inability to infer that none of the other questions is correct or to reasoning over the content of the "unanswerable" answer option, which in turn signals a strong preference for word matching. However, the model that saw only 10% of unanswerable questions in training does show a certain ability to detect them above expectations (though still unreliable). But as we saw on Figure 5, this comes at the expense of the capacity to deal with answerable questions.

Although the models trained on datasets with less uncertainty about the amount of unanswerable questions are the ones whose accuracy is better in some scenario (in the ones they were trained for), when aggregating results by model we observe that it is the models that have been trained with a sizeable amount of unanswerable questions that perform better overall on a variety of scenarios. Looking at the average accuracy over different datasets, we see that the model trained on 60% of unanswerable questions has the highest average accuracy of all models over all modified versions of *RACE*.

4.2 Metrics in terms of answerability

While general accuracy reveals that models are very susceptible to the presence of unanswerable questions in training data, in our multiple-choice setup it speaks as much about a model's capacity to determine answerability as about its ability to correctly answer the questions it deems answerable. To untwine this superimposition, we will continue examining the evaluation results by focusing on answerability alone. In this section, we dismiss the answer given to answerable questions, paying attention only to whether the system identifies unanswerable questions. What we propose is the binarization of the model's responses: instead of *A*, *B*, *C* or *D*, we will interpret the model's responses as *unanswerable* or *answerable*. A model decides a question is *unanswerable* when it chooses the option that contains *None of the answers are correct.*, and decides the question is *answerable* when it chooses any of the other 3 answers). In this way, we switch the problem from identifying the right answer to recognizing if the question is answerable given the options. Note that for the calculation of subsequent metrics, we consider *unanswerable* as the positive class.

Figure 7: Answerability accuracy, i.e. accuracy at unanswerable question detection.

We define answerability accuracy as:

$$\text{Answerability acc.} = \frac{|\text{ unanswerable} \wedge \text{pred. unanswerable }| + |\text{ answerable} \wedge \text{pred. other }|}{N}$$

On Figure 7 we observe that answerability accuracy has a distribution pattern that is similar what we saw for general accuracy. However, values towards the lower left corner of the table are higher in this case, simply indicating that models hardly ever choose the unanswerable option when it was hardly seen in training. We can see this on Figure 8:

Figure 8: Proportion of times the "unanswerable" option is chosen.

When looking at aggregates by model (the single row above the main table) on Figure 7, we see that now it is the model that saw 30% of unanswerable questions that performs best on average across all 11 testing scenarios.

We refer again to Figure 6 to observe the models' recall regarding answerability. When measuring the retrieval rate for the *unanswerable* option (i.e. recall), how we interpret *True Negatives* does not matter. If we take *unanswerable* to be the positive class, recall in terms of answerability is the same as general recall.

Figure 9: Specificity at unanswerable question detection.

The retrieval of answerable questions or *specificity* (Figure 9) yields a pattern similar to the one seen on Figure 5. Values here are generally higher, indicating that models that saw few unanswerable questions in training tend to fail by choosing "proper" but incorrect answers, not by choosing the *unanswerable* option. This again can be explained by BERT's preference for adjacent sentences that are semantically related.

4.3 Comparing results on imbalanced datasets

In this work, we compare the results of testing a series of biased models on a series of imbalanced datasets. While the datasets are (deliberately) imbalanced, we hypothesize that retrieving one class is as important as retrieving the other. In such a situation, the ideal scenario is a combination of model and test set that yields good accuracy over the two classes. But so far, the results indicate that the ability to retrieve one class is detrimental to the ability to retrieve the other. We need metric that takes into account the accuracies on each of the two classes at the same time. To that end, we propose using **Youden's J statistic** or Youden's index (Youden, 1950), defined as:

$$J = \text{recall} + \text{specificity} - 1$$

Expanding the formula we have:

$$J = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} + \frac{\text{true negatives}}{\text{true negatives} + \text{false positives}} - 1$$

Which in our case translates to:

$$J = \frac{\text{true unanswerable}}{\text{total unanswerable}} + \frac{\text{true answerable}}{\text{total answerable}} - 1$$

Youden's J statistic is a measure of informedness that gives equal weight to the two types of error: false negatives (unanswerable questions for which the system chooses a "proper" answer) and false positives (answerable questions declared unanswerable). It produces values in the range $[0 - 1]$ (by definition $[-1 - 1]$, but a negative value can be corrected by switching the classes), and can be seen as a linear transformation of the *balanced accuracy* (the arithmetic mean of recall and specificity). We have chosen Youden's J statistic over balanced accuracy because it produces a wider range of values.

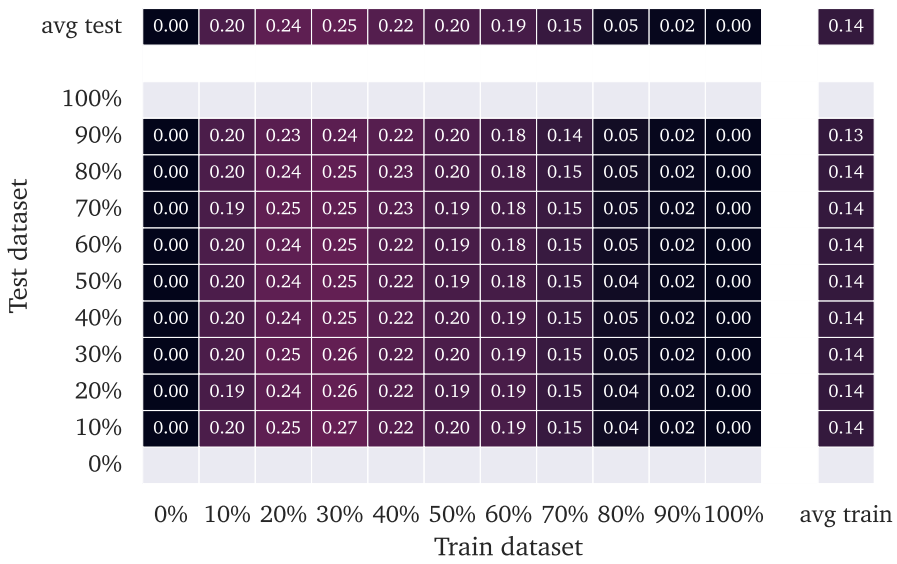

Figure 10: Youden's J statistic in terms of answerability.

The values in Figure 10 reveal that both Figure 4 and Figure 7 are too optimistic. As seen above, a model's accuracy is generally good on test sets that are similar to the one the model was trained on, which leads to good values towards the lower left and upper right corners of the tables — where train and test sets have little uncertainty concerning answerability and also match.

Here we see a different picture: as expected, the values on the leftmost and three rightmost columns are almost 0, again confirming that the respective models only have predictive capability for the class they have seen most. Results are not much better towards the centre of the table, and no value reaches 0.5, indicating all models' poor informedness concerning answerability. However, we see that the "30%" model is clearly better informed that the others.

Although our results generally speak of a big trade-off between recognizing answerability and correctly answering abilities, and do not allow us to prescript any particular training regime, the 30% of unanswerable questions in training could be an interesting focus point in combination with the proposals we make in the following, last chapter.

5 Conclusions and future work

After having looked at the evaluation results obtained from our method to find an optimal scenario for the identification of unanswerable questions in QA systems, in this chapter we draw conclusions about how well these results answer the research questions set out in Chapter 1, we assess our methodology in light of the results, and consider future directions for the present research taking note of that assessment.

5.1 Research questions revisited

Research question 1: ***What is the optimal scenario to identify unanswerable questions?*** Our experimental results indicate a strong preference for certainty regarding answerability, but not a clear path on how to deal with uncertainty regarding answerability, the main aim of the study. Models only obtained strong results when dealing with datasets that *a)* were similar to the ones they had been trained on and *b)* contained a very low or very high number of unanswerable questions. Models were for the most part unable to deal with distractors, and only reproduced training bias. Youden's J statistic and Figure 10 reveal that a proportion of 30% of unanswerable questions during training yields the most informed system, but to the general decrease in ability to correctly answer questions that might be unanswerable once they occur during training, we cannot recommend setup. Further research is needed to achieve a scenario where unanswerable questions can be recognized to a significant extent without harming the system's ability to answer answerable questions.

Research question 2: ***Does BERT need to see a higher proportion of unanswerable questions in training to be able to reliably identify them in test?*** Looking at Figure 4 by row we see that in an evaluation scenario with 50% of unanswerable questions, the amount of them seen in training does not matter as long as there were some — over 10%. For scenarios with less that 50% of unanswerable questions (presumably more likely), models that saw *less* than that amount are preferable. If we relax the criteria and look at answerability detection only (Figure 7), the evaluation scenario with 50% of unanswerable questions is also better handled by models that saw a lower proportion during test. Only for scenarios with more that 50% of unanswerable questions the results suggest indeed a higher proportion during training.

The above signals that the proportion of unanswerable questions a BERT system should see during training largely depends on the end application.

Research question 3: ***How does a trained BERT model respond when it is tested on different proportions of unanswerable questions?*** Our results show that BERT models generally benefit from a biased training. However, looking at the performance separately on each class, we see that the ability to detect answerability or to correctly answer answerable questions remains constant across

different scenarios. There is a trade-off between the two abilities which appears in any scenario, but while a model's performance depends on the evaluation scenario being biased in the same direction as the model, model's informedness stays the same. Therefore, we would advise that it is unnecessary to test models in different scenarios regarding answerability. A single scenario with 10–50% of unanswerable questions, which matches what is proposed in other literature, would suffice.

5.2 Evaluation of the proposed method and further directions

5.2.1 Adversarial methods and the multiple-choice format

Evaluation results (Figures 4, 5) clearly show how detrimental the adversarial method of answerable questions **replacement** (by unanswerable ones) was to the system's ability to effectively answer answerable questions. Given the system's almost complete inability to deal with answerable questions when their proportion in training was low, we can ascribe this inability to insufficient training data.

This points out at an alternative method consisting in the **adversarial augmentation** of the original *RACE* dataset, where modified, unanswerable questions are added along with the original, answerable ones. This would have the additional benefit of allowing further interpretation of false positives: if the system decides that a question is unanswerable while the answer is in fact present, we could study how this happened by comparing the system's response to the original question, in turn prompting the study of the original dataset and its distractor answer options.

5.2.2 Interaction of the chosen dataset format and model

As we have seen in Chapter 2, working with the multiple-choice format involves devising a format to simultaneously feed the model a passage, a question, and four answer options. Since we decided to use a baseline BERT model directly, feeding BERT these elements simultaneously means transforming a set of the form `{passage, question, 4 options}` into 4 sequences of the form `[passage, question, option]` or `[passage, question&option]` (depending on the form of the question). This has the effect of transforming answerable questions into sequences of semantically related sentences, while unanswerable questions become sequences of sentences where the last one is going to be largely unrelated to the preceding ones. At the same time, this changes the problem into selecting the most probable sequence.

BERT has a secondary pretraining objective of next sentence prediction, favouring semantically related contiguous sentences. This means BERT will give answerable and unanswerable questions very different scores, which is clearly beneficial for answerability detection. However, it does not mean that BERT is reasoning over the content of the "unanswerable" option, or that it is reasoning over the different options simultaneously. What we see in the results is that answer

options are scored independently of each other, which means that if a BERT model is biased to see more unanswerable questions, most likely it recognizes the sequence with the lowest score as the correct one, while this is not always the keys. This could be further analysed by studying the effects of different proportions of distractor questions during training and test — in the present work, we have introduced the distractor in every set of options.

5.2.3 What makes questions unanswerable?

In this work we have modified a multiple-choice QA dataset by replacing some answer options by the sentence *None of the answers are correct*. The experimental results, but also the dataset's format, make it difficult to study whether the content of the unanswerable question was ideal, since we have no data to elucidate why some unanswerable questions are detected why others are not. An opportunity to further examine this, perhaps in combination with the adversarial augmentation proposed above, would be to study the effects of different contents in the "unanswerable" option, be it special tokens, **alternative phrases** or an analysis of the system's responses when the correct answer is simply eliminated. Moreover, further inquiry into what constitutes an unanswerable question could point out at different possibilities to generate unanswerable questions from existing data.

References

Abujabal, A., Roy, R.S., Yahya, M., Weikum, G., 2018. Comqa: A community-sourced dataset for complex factoid question answering with paraphrase clusters. arXiv preprint arXiv:1809.09528.

Allam, A.M.N., Haggag, M.H., 2012. The question answering systems: A survey. International Journal of Research and Reviews in Information Sciences (IJRRIS) 2.

Back, S., Chinthakindi, S.C., Kedia, A., Lee, H., Choo, J., 2019. NeurQuRI: Neural question requirement inspector for answerability prediction in machine reading comprehension, in: International Conference on Learning Representations.

Berninger, V.W., Nagy, W., Beers, S., 2011. Child writers' construction and reconstruction of single sentences and construction of multi-sentence texts: Contributions of syntax and transcription to translation. Reading and writing 24, 151–182.

Burges, C.J., 2013. Towards the machine comprehension of text: An essay, TechReport: MSR-TR-2013-125.

Chen, D., 2018. Neural reading comprehension and beyond (PhD thesis). Stanford University.

Chen, D., Bolton, J., Manning, C.D., 2016. A thorough examination of the cnn/daily mail reading comprehension task. arXiv preprint arXiv:1606.02858.

Clark, C., Gardner, M., 2017. Simple and effective multi-paragraph reading comprehension. arXiv preprint arXiv:1710.10723.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., Toutanova, K., 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. arXiv preprint arXiv:1905.10044.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., Tafjord, O., 2018. Think you have solved question answering? Try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457.

Dai, A.M., Le, Q.V., 2015. Semi-supervised sequence learning. Advances in neural information processing systems 28.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R., 2019. Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860.

Dasigi, P., Liu, N.F., Marasović, A., Smith, N.A., Gardner, M., 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. arXiv preprint arXiv:1908.05803.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R., 1990. Indexing by latent semantic analysis. Journal of the American society for information science 41, 391–407.

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

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., Gardner, M., 2019. DROP: A reading comprehension benchmark requiring discrete reasoning

over paragraphs. arXiv preprint arXiv:1903.00161.
Erhan, D., Courville, A., Bengio, Y., Vincent, P., 2010. Why does unsupervised pre-training help deep learning?, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop; Conference Proceedings, pp. 201–208.
Furnas, G.W., Landauer, T.K., Gomez, L.M., Dumais, S.T., 1987. The vocabulary problem in human-system communication. Communications of the ACM 30, 964–971.
Gan, W.C., Ng, H.T., 2019. Improving the robustness of question answering systems to question paraphrasing, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 6065–6075.
Gardner, M., Berant, J., Hajishirzi, H., Talmor, A., Min, S., 2019. Question answering is a format; when is it useful? arXiv preprint arXiv:1909.11291.
Green Jr, B.F., Wolf, A.K., Chomsky, C., Laughery, K., 1961. Baseball: An automatic question-answerer, in: Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference. pp. 219–224.
Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P., 2015. Teaching machines to read and comprehend. Advances in neural information processing systems 28.
Hirschman, L., Light, M., Breck, E., Burger, J.D., 1999. Deep read: A reading comprehension system, in: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics. pp. 325–332.
Howard, J., Ruder, S., 2018. Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146.
Hu, M., Wei, F., Peng, Y., Huang, Z., Yang, N., Li, D., 2019. Read+verify: Machine reading comprehension with unanswerable questions, in: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 6529–6537.
Jia, R., Liang, P., 2017. Adversarial examples for evaluating reading comprehension systems. arXiv preprint arXiv:1707.07328.
Jones, K.S., 1972. A statistical interpretation of term specificity and its application in retrieval. Journal of documentation.
Joshi, M., Choi, E., Weld, D.S., Zettlemoyer, L., 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. arXiv preprint arXiv:1705.03551.
Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S., Roth, D., 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 252–262.
Kočiskỳ, T., Schwarz, J., Blunsom, P., Dyer, C., Hermann, K.M., Melis, G., Grefenstette, E., 2018. The narrativeqa reading comprehension challenge. Transactions of the Association for Computational Linguistics 6, 317–328.
Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., others, 2019. Natural questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics 7, 453–466.

Lai, G., Xie, Q., Liu, H., Yang, Y., Hovy, E., 2017. Race: Large-scale reading comprehension dataset from examinations. arXiv preprint arXiv:1704.04683.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R., 2019. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

Levesque, H., Davis, E., Morgenstern, L., 2012. The winograd schema challenge, in: Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning.

Levesque, H.J., 2014. On our best behaviour. Artificial Intelligence 212, 27–35.

Lhoest, Q., Moral, A.V. del, Jernite, Y., Thakur, A., Platen, P. von, Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., others, 2021. Datasets: A community library for natural language processing. arXiv preprint arXiv:2109.02846.

Liu, S., Zhang, X., Zhang, S., Wang, H., Zhang, W., 2019. Neural machine reading comprehension: Methods and trends. Applied Sciences 9, 3698.

Liu, X., Shen, Y., Duh, K., Gao, J., 2017. Stochastic answer networks for machine reading comprehension. arXiv preprint arXiv:1712.03556.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L., 2016. MS MARCO: A human generated machine reading comprehension dataset, in: CoCo@ NIPS.

Norvig, P., 1986. Unified theory of inference for text understanding (PhD thesis). CALIFORNIA UNIV BERKELEY GRADUATE DIV.

Onishi, T., Wang, H., Bansal, M., Gimpel, K., McAllester, D., 2016. Who did what: A large-scale person-centered cloze dataset. arXiv preprint arXiv:1608.05457.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., others, 2019. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems 32.

Peñas, A., Hovy, E., Forner, P., Rodrigo, Á., Sutcliffe, R., Forăscu, C., Sporleder, C., 2012. Evaluating machine reading systems through comprehension tests, in: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12). pp. 1143–1147.

Pradel, C., Sileo, D., Rodrigo, Á., Peñas, A., Agirre, E., 2020. Question answering when knowledge bases are incomplete, in: International Conference of the Cross-Language Evaluation Forum for European Languages. Springer, pp. 43–54.

Qiu, B., Chen, X., Xu, J., Sun, Y., 2019. A survey on neural machine reading comprehension. arXiv preprint arXiv:1906.03824.

Rajpurkar, P., Jia, R., Liang, P., 2018. Know what you don't know: Unanswerable questions for SQuAD. arXiv preprint arXiv:1806.03822.

Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P., 2016. Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250.

Richardson, M., Burges, C.J., Renshaw, E., 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text, in: Proceedings of the

2013 Conference on Empirical Methods in Natural Language Processing. pp. 193–203.

Roberts, A., Raffel, C., Shazeer, N., 2020. How much knowledge can you pack into the parameters of a language model? arXiv preprint arXiv:2002.08910.

Rodrigo, A., Penas, A., 2017. A study about the future evaluation of question-answering systems. Knowledge-Based Systems 137, 83–93.

Rodrigo, A., Penas, A., Miyao, Y., Hovy, E.H., Kando, N., 2015. Overview of CLEF QA entrance exams task 2015. CLEF (Working Notes) 56, 59–99.

Rogers, A., Gardner, M., Augenstein, I., 2021. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. arXiv preprint arXiv:2107.12708.

Rogers, A., Kovaleva, O., Downey, M., Rumshisky, A., 2020. Getting closer to AI complete question answering: A set of prerequisite real tasks, in: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 8722–8731.

Seo, M., Kembhavi, A., Farhadi, A., Hajishirzi, H., 2016. Bidirectional attention flow for machine comprehension. arXiv preprint arXiv:1611.01603.

Simmons, R.F., Klein, S., McConlogue, K., 1964. Indexing and dependency logic for answering english questions. American Documentation 15, 196–204.

Sugawara, S., Inui, K., Sekine, S., Aizawa, A., 2018. What makes reading comprehension questions easier? arXiv preprint arXiv:1808.09384.

Sun, F., Li, L., Qiu, X., Liu, Y., 2018. U-net: Machine reading comprehension with unanswerable questions. arXiv preprint arXiv:1810.06638.

Tan, C., Wei, F., Zhou, Q., Yang, N., Lv, W., Zhou, M., 2018. I know there is no answer: Modeling answer validation for machine reading comprehension, in: CCF International Conference on Natural Language Processing and Chinese Computing. Springer, pp. 85–97.

Tang, G., Müller, M., Rios, A., Sennrich, R., 2018. Why self-attention? A targeted evaluation of neural machine translation architectures. arXiv preprint arXiv:1808.08946.

Tenney, I., Das, D., Pavlick, E., 2019. BERT rediscovers the classical NLP pipeline. arXiv preprint arXiv:1905.05950.

Turing, A.M., 1950. Computing machinery and intelli- gence. Mind LIX, 433–460. https://doi.org/10.1093/mind/LIX.236.433

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems 30.

Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A., 2008. Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning. pp. 1096–1103.

Wang, W., Yang, N., Wei, F., Chang, B., Zhou, M., 2017. Gated self-matching networks for reading comprehension and question answering, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 189–198.

Wang, Y., Bansal, M., 2018. Robust machine comprehension models via adversarial training, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Lan-

guage Technologies, Volume 2 (Short Papers). Association for Computational Linguistics, New Orleans, Louisiana, pp. 575–581. https://doi.org/10.18653/v1/N18-2091

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., others, 2019. Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

Woods, W., 1972. The lunar sciences natural language information system. BBN report.

Xie, Q., Lai, G., Dai, Z., Hovy, E., 2017. Large-scale cloze test dataset created by teachers. arXiv preprint arXiv:1711.03225.

Yang, Ziqing, Cui, Y., Che, W., Liu, T., Wang, S., Hu, G., 2019. Improving machine reading comprehension via adversarial training. arXiv preprint arXiv:1911.03614.

Yang, Zhilin, Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V., 2019. Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems 32.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W.W., Salakhutdinov, R., Manning, C.D., 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. arXiv preprint arXiv:1809.09600.

Youden, W.J., 1950. Index for rating diagnostic tests. Cancer 3, 32–35.

Yun, C., Bhojanapalli, S., Rawat, A.S., Reddi, S.J., Kumar, S., 2019. Are transformers universal approximators of sequence-to-sequence functions? arXiv preprint arXiv:1912.10077.

Zhang, X., Yang, A., Li, S., Wang, Y., 2019. Machine reading comprehension: A literature review. arXiv preprint arXiv:1907.01686.

Zhang, Z., Wu, Y., Zhao, H., Li, Z., Zhang, S., Zhou, X., Zhou, X., 2020a. Semantics-aware BERT for language understanding, in: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 9628–9635.

Zhang, Z., Wu, Y., Zhou, J., Duan, S., Zhao, H., Wang, R., 2020b. SG-net: Syntax-guided machine reading comprehension, in: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 9636–9643.

Zhang, Z., Yang, J., Zhao, H., 2020c. Retrospective reader for machine reading comprehension. arXiv preprint arXiv:2001.09694 1, 1–9.

Zhang, Z., Zhao, H., Wang, R., 2020d. Machine reading comprehension: The role of contextualized language models and beyond. arXiv preprint arXiv:2005.06249.

Zhu, H., Dong, L., Wei, F., Wang, W., Qin, B., Liu, T., 2019. Learning to ask unanswerable questions for machine reading comprehension. arXiv preprint arXiv:1906.06045.

A Appendix: Experimental results by difficulty level

A.1 Experimental results on *RACE high*

Figure 11: Model accuracy on modified *RACE high* test sets

Figure 12: Model accuracy on modified *RACE high* test sets when taking only answerable questions into account.

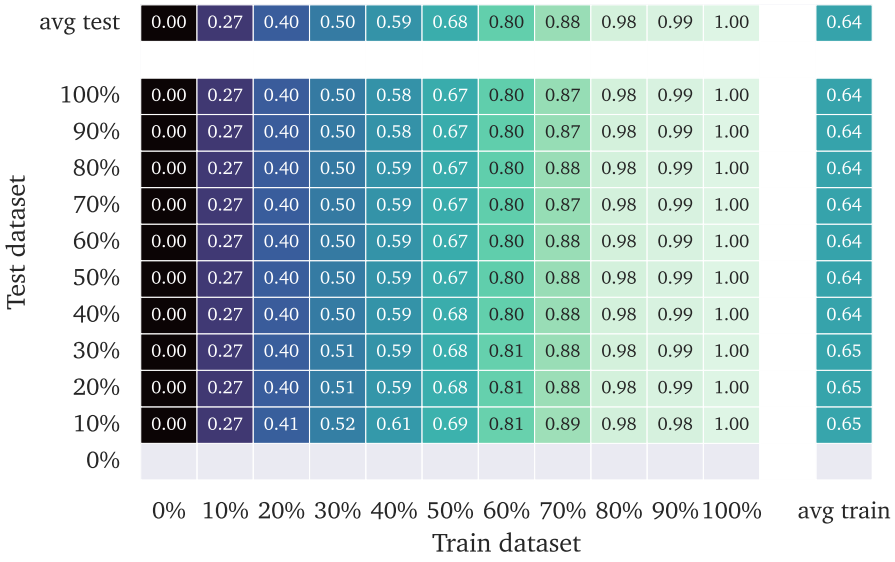

Figure 13: Accuracy on unanswerable questions only, or recall, on modified *RACE high* test sets.

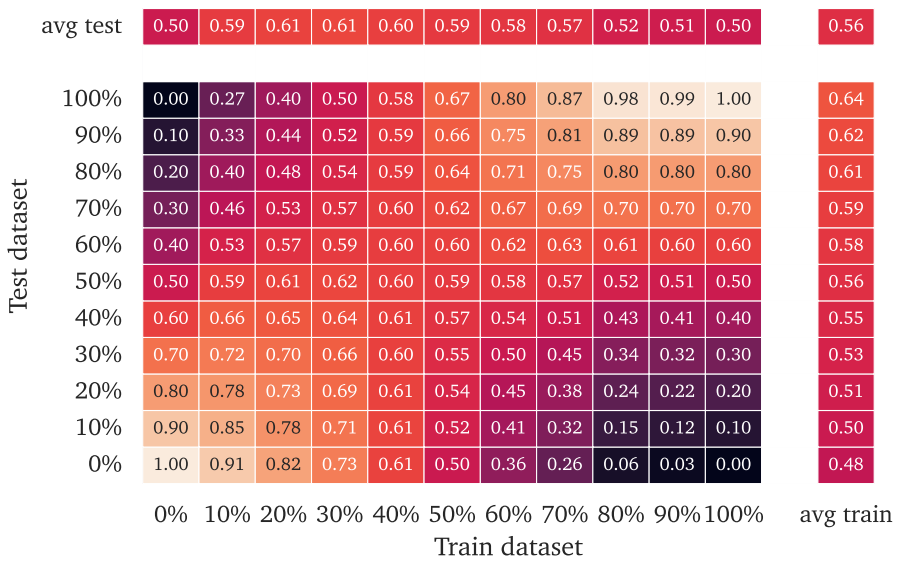

Figure 14: Accuracy at unanswerable question detection on modified *RACE high* test sets.

Figure 15: Proportion of times the "unanswerable" option is chosen on modified *RACE high* test sets.

Figure 16: Specificity at unanswerable question detection on modified *RACE high* test sets.

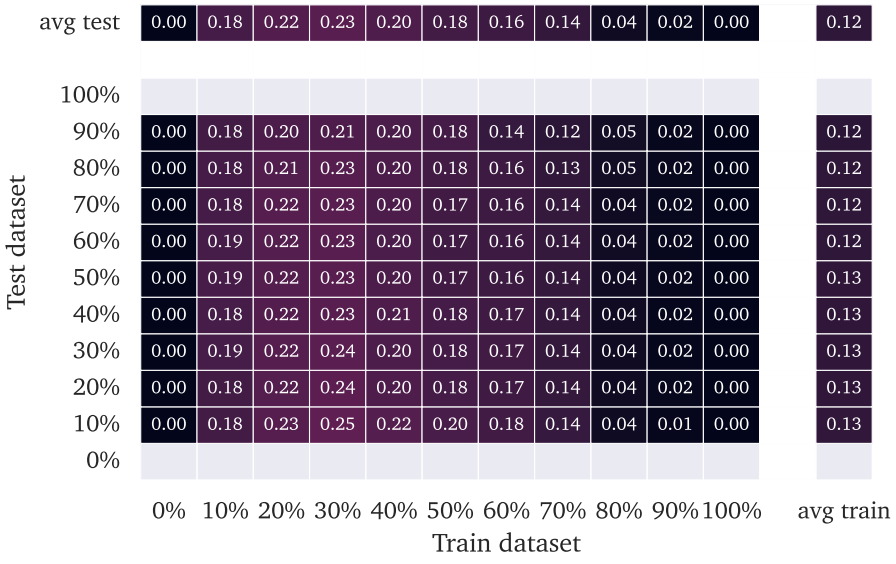

Figure 17: Youden's J statistic in terms of answerability on modified *RACE high* test sets.

A.2 Experimental results on *RACE middle*

Figure 18: Model accuracy on modified *RACE middle* test sets

Figure 19: Model accuracy on modified *RACE middle* test sets when taking only answerable questions into account.

Figure 20: Accuracy on unanswerable questions only, or recall, on modified *RACE middle* test sets.

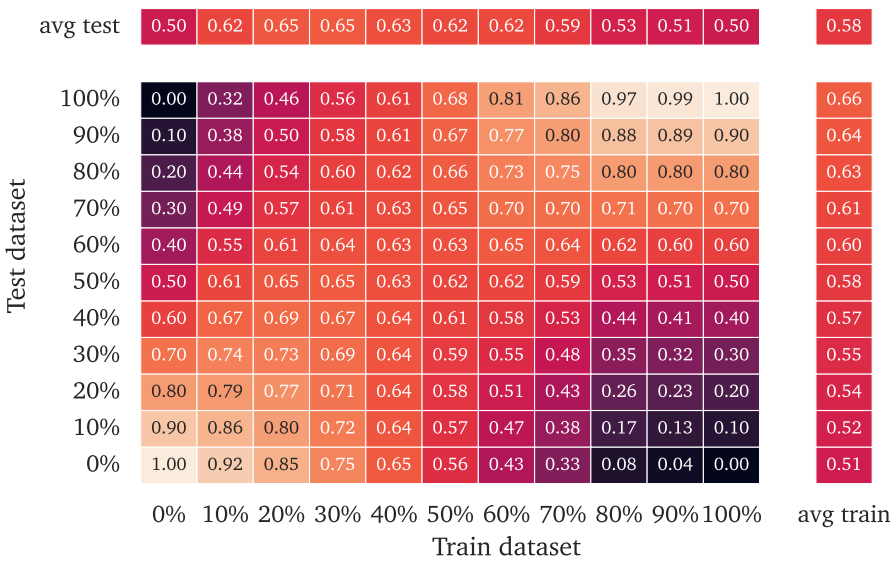

Figure 21: Accuracy at unanswerable question detection on modified *RACE middle* test sets.

Figure 22: Proportion of times the "unanswerable" option is chosen on modified *RACE middle* test sets.

Figure 23: Specificity at unanswerable question detection on modified *RACE middle* test sets.

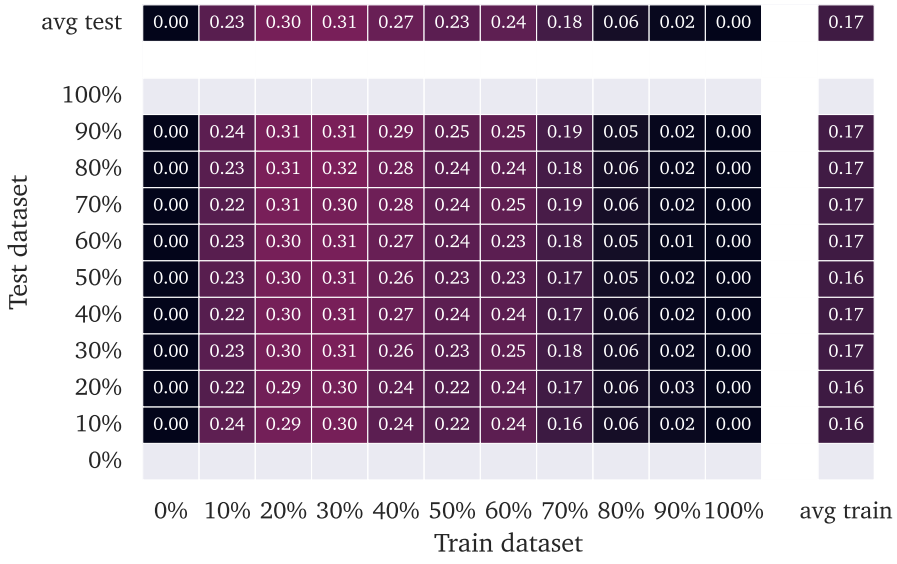

Figure 24: Youden's J statistic in terms of answerability on modified *RACE middle* test sets.