English Machine Reading Comprehension: New Approaches to Answering Multiple-choice Questions

Daria Dzendzik

Diploma in Mathematics and Software Engineering

A dissertation submitted in fulfilment of the requirements for the award of

Doctor of Philosophy (Ph.D.)

to the

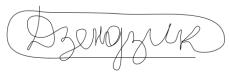
Dublin City University
School of Computing

Supervisors:
Dr. Jennifer Foster
Prof. Carl Vogel (Trinity College Dublin, Ireland)

2022

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D. is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:



(Candidate) ID No.: 16210692

Date: 28.06.2021

To my mom, my dad, and two Andrews.

Посвящатся моей маме, отцу и двум Андреям.

Contents

List of Figures

List of Tables

English Machine Reading Comprehension: A Systematic Overview, and New Approaches to Answering Multiple-choice and Boolean Questions

Daria Dzendzik

Abstract

Reading comprehension is often tested by measuring a person or system's ability to answer questions about a given text. Machine reading comprehension datasets have proliferated in recent years, particularly for the English language. The aim of this thesis is to investigate and improve data-driven approaches to automatic reading comprehension.

Firstly, I provide a full classification of question and answer types for the reading comprehension task. I also present a systematic overview of English reading comprehension datasets (over 50 datasets). I observe that the majority of questions were created using crowdsourcing and the most popular data source is Wikipedia. There is also a lack of *why*, *when*, and *where* questions. Additionally, I address the question "*What makes a dataset difficult?*" and highlight the difference between datasets created for people and datasets created for machine reading comprehension. Secondly, focusing on multiple-choice question answering, I propose a computationally light method for answer selection based on string similarities and logistic regression. At the time (December 2017), the proposed approach showed the best performance on two datasets (MovieQA and MCQA: IJCNLP 2017 Shared Task 5 Multi-choice Question Answering in Examinations) outperforming some CNN-based methods. Thirdly, I investigate methods for Boolean Reading Comprehension tasks including the use of Knowledge Graph (KG) information for answering questions. I provide an error analysis of a transformer model's performance on the BoolQ dataset. This reveals several important issues such as unstable model behaviour and some issues with the dataset itself. Experiments with incorporating knowledge graph information into a baseline transformer model do not show a clear improvement due to a combination of the model's ability to capture new information, inaccuracies in the knowledge graph, and imprecision in entity linking. Finally, I develop a Boolean Reading Comprehension dataset based on spontaneously user-generated questions and reviews which is extremely close to a real-life question-answering scenario. I provide a classification of question difficulty and establish a transformer-based baseline for the new proposed dataset.

Publications

The initial idea of this work was inspired by a paper the author of this thesis also co-authored but the content of which is not included in this thesis:

> Dasha Bogdanova, Jennifer Foster, **Daria Dzendzik**, and Qun Liu, (2017). If you can't beat them join them: Handcrafted features complement neural nets for non-factoid answer reranking. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 121–131, Valencia, Spain

Large portions of Chapter 2 have appeared in the following paper:

> **Daria Dzendzik**, Carl Vogel, and Jennifer Foster (2020) English Machine Reading Comprehension Datasets: A Survey In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 8784–8804, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics

Large portions of Chapter 4 have appeared in the following papers:

> **Daria Dzendzik**, Carl Vogel and Qun Liu. (2017) Who Framed Roger Rabbit? Multiple Choice Questions Answering about Movie Plot The Joint Video and Language Understanding Workshop: MovieQA and The Large Scale Movie Description Challenge (LSMDC), at ICCV 2017, 23th of October, Venice, Italy
>
> **Daria Dzendzik**, Alberto Poncelas, Carl Vogel and Qun Liu. (2017) ADAPT centre cone team at IJCNLP-2017 task 5: A Similarity-based Logistic Regression Approach to Multi-choice Question Answering in an Examinations Shared Task. In C.-H. Liu, P. Nakov, and N. Xue, editors, Proceedings of the IJCNLP 2017, Shared Tasks, page 67–72. Asian Federation of Natural Language Processing.

Large portions of Chapter 5 and Chapter 6 have appeared in the following paper:

> **Daria Dzendzik**, Carl Vogel, and Jennifer Foster. (2020). Q. Can knowledge graphs be used to answer Boolean questions? A. It's complicated! In Proceedings of the First Workshop on Insights from Negative Results in NLP, pages 6–14, Online. Association for Computational Linguistics.

Large portions of Chapter 7 have appeared in the following paper:

> **Daria Dzendzik**, Carl Vogel, and Jennifer Foster. (2019) Is it dish washer safe? Automatically answering "yes/no" questions using customer reviews. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 1–6, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

Another publication that was co-authored during this PhD but is not directly related to the work:

> Piyush Arora, Dimitar Shterionov, Yasufumi Moriya, Abhishek Kaushik, **Daria Dzendzik**, Gareth Jones. (2020) An Investigative Study of Multi-Modal Cross-Lingual Retrieval Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)

Acknowledgments

First of all, I would like to thank my both supervisors Jennifer Foster and Carl Vogel for being the best supervisors I could have asked for, for all scientific and emotional support they gave me during years of my Ph.D. journey. I would also like to thank Qun Liu for his supervision during my first two years of Ph.D., his valuable advice that helped me. I am very grateful to my examiners Walter Daelemans and Annalina Caputo for their valuable comments and making my viva experience very nice.

My internship at Google Research Switzerland in 2019 contributed into my research and knowledge. A big thank you to Massimo Nicosia and Yasemin Altun whom I was very lucky to have as a mentor.

A very special thank I would like to send to my fellow and good friend Dasha Bogdanova who become as a sister to me. She is not only inspired me a lot scientifically, professionally, and personally, but helped practically and emotionally in the beginning of my journey. Dasha is definitely responsible for my coffee addiction. I would like to thank my dear friend Valeria Filimonova who find a time to visit me no matter where I was, and spend hours talking to me during early morning times or lunch breaks. Huge thank you to my friends Lera, Kolya, Dasha, Vita, Jenya, Marina, and Nastya for supporting me and just being there for me physically and distantly when I needed. Incredible thank you to the best flatmates ever: Andrea, Anderson and Sean (Shinan). You made the work-from-home experience during the pandemic so much better.

I would also like to thank my fellows Alberto, Eva, Dimi, Ke, Koel, Yasu, James, Henry, Meghan, and Abigail for creating a friendly atmosphere in the lab, for many great discussions we had. Thanks to Graziano D'Innocenzo, Darragh Blake, Cormac McKenna, and Joachim Wagner for doing a great job maintaining the cluster where I ran most of my experiments. I am also very thankful to Laura Grehan, Colm O'Hehir, Cara Green, Emma Clarke, and Jane Dunne for enormous impact of social and professional aspects of my life during my Ph.D.

I would like to also thank my friends Maksim Tkachenko, Elena Sivogolovko, Neel Mani and Amy Siu who supported and inspired me during my work and long time before that.

I am also deeply grateful to Boris Novikov, Sergey Serebryakov, Natalia Vassillieva and Anton Bondarev who had been my the first advisors I had back at St.Petersburg State University and Hewlett Packard Labs. Without them, I would not be where I am.

I want to thank my mother and father for caring so much for me, and for having done their best to give me a good education. And also my younger brother who managed to get his Ph.D. back in 2016. Also big thank you for my extended Ukrainian and Irish families. Your support really helped me all this time.

Finally, a big thank you to Andrew who survived being next to me while I was working on my thesis. Thank you for proofreading my papers and reports, and for spending our first St. Valentine Day in the lab next to me due to paper deadline, for being so patient and kind to me all this time.

Chapter 1

Introduction

> The more that you read, the more things you will know. The more that you learn, the more places you'll go.
>
> *Dr. Seuss*
>
> *I Can Read With My Eyes Shut!*

It is a part of human nature to be curious and ask questions, and it is natural for a person to look for an answer by asking around or by reading. Every day a vast amount of textual data is generated and made available such as news reports, articles, blog-posts, reviews and other documents. This textual data contains the answer to a myriad of questions.

Reading comprehension (RC) is the ability to read a text, process it, and understand its meaning. This task is widely used for educational purposes. International examinations such as Graduate Record Examinations (GRE)[1] and International English Language Testing System (IELTS)[2] contain different types of tasks to measure this ability, including asking questions about provided text. Davis (1944) provided nine groups of skills which are particularly important in reading, including knowledge of word meanings, ability to answer questions that are specifically answered in a passage, ability to answer questions

[1] `https://www.ets.org/gre` – last verified June 2021
[2] `https://www.ielts.org/` – last verified December 2021

that are answered in a passage but not in the words in which the question is asked, and ability to draw inferences from a passage about its contents.

The task of teaching machines to "understand" natural human language is one of the key challenges in Artificial Intelligence (AI). And in the same way that human understanding can be tested with reading comprehension tests, the ability of machines to process language can be tested in the same way. Machine Reading Comprehension (MRC) is not an abstract task of AI, as it has a lot of practical applications, including, but not limited to:

- satisfying a general user's curiosity, as a lot of questions addressed to the search engine, and the answer might be found in an encyclopedia page, user forum, etc.;
- searching for an answer for a domain specific question from a document or collection of documents;
- implementation of chat-bots which are able to answer user queries about a particular product and service within a specified context.

1.1 Reading Comprehension and Question Answering

As there are particular tests used to evaluate human reading comprehension, there are different tests (datasets) designed for machines. Based on available datasets, I categorise machine reading comprehension into three sub-tasks: *Inference*, *Text Completion*, and *Question Answering*.

Inference or **recognizing textual entailment** is a task to determine if one statement can be concluded from another statement. In other words, whether two texts are entailments, contradictions, or neutral to each other. The inference task was studied by many, including Fyodorov et al. (2000, 2003); Condoravdi et al. (2003); Bos and Markert (2005); Dagan et al. (2006); MacCartney and Manning (2007); Williams et al. (2018). The dataset introduced by Williams et al. (2018) became a part of the GLUE Benchmark[3] described

[3] `https://gluebenchmark.com/` – last verified June 2021

by Wang et al. (2018).

Text Completion is the task of completing a story in the most coherent way. It can be a missing word, entire sentence, or a larger part of text which might be generated or selected from provided options. The text can be unfinished and the task is to finish it (Zellers et al., 2018, 2019), or the text might have missing words and those gaps should be filled (Mostafazadeh et al., 2016, 2017; Xie et al., 2018).

Text completion is consistent with ability to recognize the literary devices used in a passage and to determine its tone, mood, and the writer's purpose, intent, and point. So, according to Davis (1944), both inference and text completion reflect skills which are deemed to be important for reading comprehension.

And finally, **Question Answering** (in the context of MRC) is a task of answering questions based on provided text or a set of documents. This is the type of reading comprehension I focus on in my thesis: the automatic reading comprehension question answering task. The rest of this thesis will describe this task in detail.

Question answering (QA) in general is one of the most studied tasks in Natural Language Processing (NLP). The Text Retrieval Evaluation Conference (TREC)[4] had a question answering track since 1999. Since then the community has been paying attention to this task in different forms such as open/closed domain, factoid/non-factoid, common sense, community question answering, conversational QA, visual QA, and multi-modal QA. Generally speaking, question answering is not limited to reading comprehension but one of the approaches to evaluate reading comprehension is via question answering.

In other words, the focus of this work is the intersection of reading comprehension and question answering: Machine Reading Comprehension Question Answering – see Figure 1.1. I focused on this setting as it is the closest to an application that requires machine reading comprehension.

Formally, I follow the definition of the MRC QA task as a supervised learning problem proposed by Chen (2018) and presented in Definition 1.1:

[4] `http://trec.nist.gov/` - last verified June 2021

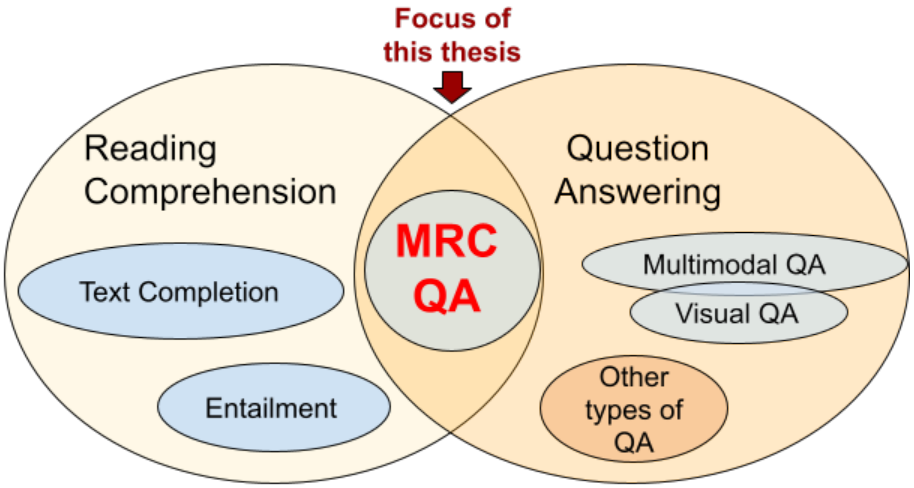


Figure 1.1: Machine Reading Comprehension and Question Answering.

Definition 1.1. Given a collection of training examples $\{(p_i, q_i, a_i)\}_{i=1}^{n}$, the goal is to learn a predictor f which takes a passage of text p and a corresponding question q as inputs and gives the answer a as output.

$f : (p, q) \rightarrow a$

Now that the general area of the work has been established, let us move to the specific research questions.

1.2 Research Questions

The majority of methods used for reading comprehension are based on neural networks where the features are opaque to humans. A central goal of this thesis is to better understand the specific challenges of reading comprehension and improve the state-of-the-art approaches to question answering by combining machine learning techniques with a variety of feature types, including string similarity and knowledge graph information.

The first research question is:

RQ 1: What kind of reading comprehension datasets are available, what kind of tasks do they cover, and what makes a dataset difficult? I provide a full classification of question and answer types for the reading comprehension task and present a systematic overview of English reading comprehension datasets, including statistics, vocabulary, and named entity analysis. I also answer the following sub-questions:

- What kind of data is used for creating those datasets?
- How do they overlap?
- What is the most common type of question?
- Are there particular types of questions which are underrepresented in existing datasets?
- What makes a dataset difficult?
- How is a reading comprehension dataset created for people different from a dataset created for machine reading comprehension?

Considering the Reading Comprehension Question Answering task more specifically, the second research question can be formulated as follows:

RQ 2: How can a combination of features based on the similarity of natural language representations impact the state-of-the-art results in a multiple-choice reading comprehension task?

I explore how different types of similarities can be combined together to achieve and outperform the state-of-the-art performance on two datasets: MovieQA (Tapaswi et al., 2016) and MCQA (Shangmin et al., 2017).

Focusing on boolean questions, the third research question can be formulated as follows:

RQ 3: How can knowledge graph information be used to impact the state-of-the-art methods in the boolean reading comprehension task?

I explore the challenges of boolean question answering and experiment with possible combinations of additional knowledge resources in a reading comprehension task with boolean questions. I investigate the following sub-questions:

- Does a reading comprehension model benefit from adding textual data obtained from a knowledge base?

- Does the addition of structured information about entities in the question and passage, and the relations between them help a reading comprehension model to answer questions?

Finally, the last research question can be defined as:

RQ 4: How do state-of-the-art MRC approaches perform with user generated data in the boolean reading comprehension task?

Specifically, I investigate how state-of-the-art approaches perform on questions and passages which were formulated without the intention of being used in any sort of MRC datasets.

1.3 Contributions

The contributions of this thesis are:

1. **Systematic overview of over 50 English MRC datasets**

   I provide a full classification of question and answer types for the reading comprehension task. I also present a systematic overview of over 50 English reading comprehension datasets. I observe that the majority of questions were created using crowdsourcing and the most popular data source is Wikipedia. There is also a lack of *Why*, *When*, and *Where* questions. Majority of questions (and answers) across different domains contains *Person* named entity in it. I address the question *What makes a dataset difficult?* and highlight the difference between datasets created for people and datasets created for machine reading comprehension.

2. **A computationally light state-of-the-art (at the time) approach for multiple choice MRC**

   Focusing on multiple-choice question answering, I propose a computationally light method for multiple choice MRC based on string similarities and logistic regression. At the time (December 2017), the proposed approach showed the best performance on two datasets (MovieQA (Tapaswi et al., 2016) and MCQA: IJCNLP

2017 Shared Task5 Multi-choice Question Answering in Examinations (Shangmin et al., 2017)) outperforming some CNN-based methods.

3. **Boolean question analysis and investigation of usage knowledge graphs in MRC**
   I investigate the methods for Boolean Reading Comprehension tasks including the use of Knowledge Graph (KG) information for answering questions. I provide an error analysis of a transformer model's performance on the BoolQ dataset (Clark et al., 2019) that reveals several important issues such as unstable model behaviour and some issues with the dataset itself. Experiments with incorporating knowledge graph information into a baseline transformer model do not show a clear improvement due to a combination of the model's ability to capture new information, inaccuracies in the knowledge graph, and imprecision in entity linking.

4. **Developed a user-generated boolean MRC dataset**
   I develop a Boolean Reading Comprehension dataset based on spontaneously user-generated questions and reviews which is extremely close to a real-life question-answering scenario. I provide a classification of question difficulty and establish a transformer-based baseline for the new dataset. I investigate a number of passage options and experiment with a number of transformer-based architectures and transfer learning.

1.4 Thesis Structure

This thesis is structured as follows:

Chapter 2

I present the overview of reading comprehension question answering datasets. First, I discuss other datasets surveys (Zhang et al., 2019; Liu et al., 2019a; Qiu et al., 2019a; Baradaran et al., 2020; Wang, 2020; Dunietz et al., 2020; Zeng et al., 2020) and explain why my survey is different. I then categorise those datasets into eight types according to question, answer, and passage form. I compare the main features of the data such as:

domain, passage and question sources, creation methods, basic statistics, question words, vocabulary, named entities, and complexity, including human performance.

Chapter 3

I review the background of information retrieval and artificial neural networks. In particular, first, I briefly discuss word embedding representations categorised into three classes: *Frequency-based*, *Static*, and *Contextual*. Second, I introduce string similarities. Third, I discuss logistic regression. Then, I discuss the common NLP neural network architectures, such as recurrent neural networks, including Long Short Term Memory networks (Hochreiter and Schmidhuber, 1997), attention mechanism (Hermann et al., 2015), followed by memory networks (Weston et al., 2015). Then, I talk about the transformer architecture (Vaswani et al., 2017). Finally, I talk about graph-based neural networks (Scarselli et al., 2009) and their use in transformers. I explain how all those approaches work and are used in reading comprehension.

Chapter 4

I introduce an approach to the multiple choice reading comprehension task. I address this task by using a logistic regression over number of string similarities. I experimentally show that this approach provides good results on two datasets: MovieQA (Tapaswi et al., 2016) and IJCNLP-2017 Shared Task 5 (Shangmin et al., 2017). I also investigate the impact of the particular similarity features on the performance of this method by using an ablation study.

Chapter 5

I present the task of boolean question answering in reading comprehension. I start by providing the motivation on why it is worth studying boolean questions. Then I examine the BoolQ (Clark et al., 2019) dataset, analysing the reprodicibility of the baseline system and error analysis.

Chapter 6

I investigate whether a knowledge graph can be useful in the boolean reading comprehension task. First I give an overview of knowledge graphs and provide a literature review on the topic. Then I investigate possible ways of inserting knowledge graph information into

the passage and the model. I present two approaches to combining text with knowledge graph information: transforming KG triples into sentences which are added to the passage and the use of a neural network with a graph component based on the work of Shaw et al. (2019).

Chapter 7

I introduce a new boolean dataset, AmazonYesNo, based on customer questions and product reviews, and inspect different categories of questions. I investigate approaches to answering user generated questions based on spontaneous user generated content. I set up a transformer baseline and investigate the performance on user-provided answers. I look into different passage options for this task. I introduce the setup I used in my experiments and discuss the results and errors.

Chapter 8

I summarize the outcome of the thesis, list some questions that remain unsolved, and suggest directions for future work.

Chapter 2

Reading Comprehension Datasets

> It is a capital mistake to theorize before one has data.
>
> *Sir Arthur Conan Doyle*
>
> *Sherlock Holmes*

Machine reading comprehension datasets have proliferated in recent years, particularly for the English language – see Figure 2.1. The aim of this chapter is to make sense of the landscape by providing, as extensive as possible, a survey of English machine reading comprehension (MRC) datasets.

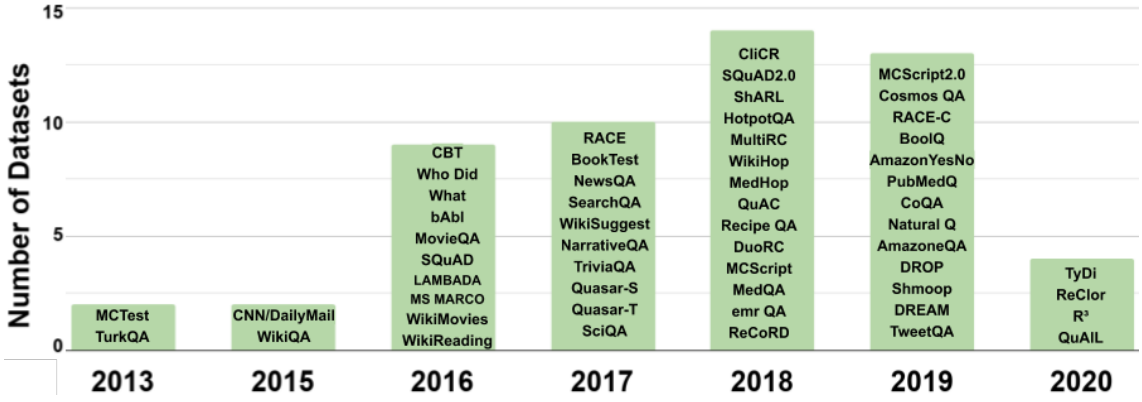


Figure 2.1: English MRC datasets released per year

To better understand the task itself as well as understanding a reason for the success and failure of particular reading comprehension methods, I present a complete overview of reading comprehension datasets including an analysis of data sources, question words,

vocabulary, and named entities.

The materials presented in this chapter will be useful to:

1. those who are new to the field and would like to get a quick yet informative overview of English MRC datasets;

2. those who are planning to create a new MRC dataset;

3. those who are interested in transfer or joint learning for MRC and are looking for potentially compatible datasets.

The chapter is structured as follows: I start with a brief description of other surveys in Section 2.1. In Section 2.2, I describe types of reading comprehension question answering based on question and answer form. Section 2.3 contains a table with the datasets and a discussion of the main characteristics such as data sources and method of creation. In Section 2.4, I provide a quantitative analysis including an analysis of question words, and vocabulary. I examine the prevalence of named entities in Section 2.5. In Section 2.6, I discuss human performance. I address difficulty of MRC datasets in Section 2.7. The chapter concludes with a summary in Section 2.8.

2.1 Related Work

A number of similar surveys have been carried out previously but mine differs in its breadth – 54 datasets compared to the two next largest, 47 (Zeng et al., 2020) and 29 (Baradaran et al., 2020) – and its focus on MRC *datasets* rather than on MRC *systems*. My survey takes a mostly structured form, with the following information presented for each dataset: size, data source, creation method, human performance level (HP) and whether the dataset has been "solved", availability of a leaderboard, the most frequent first question word, and whether the dataset is publicly available. I also categorise each dataset by its question/answer type.

The survey closest to mine in its coverage is Zeng et al. (2020). They report only the train/dev/test sizes of each dataset, whereas I calculate the length of each question,

passage and answer.[1] I examine the vocabulary of each dataset, looking at question words and applying language identification tools, and discover question inbalance and noise (HTML tags, misspelling, etc.). I also report data re-use across datasets.

Table 2.1 contains a comprehensive comparison of my work with other surveys:

Language: Zhang et al. (2019); Baradaran et al. (2020); Wang et al. (2020a) include in their surveys not only English but also Chinese datasets, while other surveys including mine considered only English datasets.

Task and Question/Answer/Paragraph Classification: Following Chen (2018), it is a common trend to divide datasets by tasks such as: cloze, multiple choice, extractive (or span extraction), and generative (abstractive or free form answers). The classification of all three (passages, questions, and answers) is presented only in two others surveys: Zeng et al. (2020) and Baradaran et al. (2020). The latter one categorize questions as factoid, non-factoid, and yes/no questions; context as multi-passage and single-passage; and answers additionally as quiz and detail.[2]

Liu et al. (2019a) define MRC specifically as a QA about text. They reviewing MRC methods including a detailed review of 15 datasets with examples. Zhang et al. (2019) also looked into new tendencies such as using knowledge base data, adding unanswerable questions, conversational datasets, and dataset with multiple passages bringing the total number of reviewed datasets to 23. Baradaran et al. (2020) conduct a review of MRC methods and analysis of related papers mentioning 29 datasets in English and Chinese created from 2016 to 2018.

Additional Analysis: Only 3 out of 8 other surveys (Zhang et al., 2019; Baradaran et al., 2020; Zeng et al., 2020) briefly mention domain while I consider it in more detail. Three other surveys (Liu et al., 2019a; Baradaran et al., 2020; Zeng et al., 2020) discuss metrics for MRC evaluation but none of those pay attention to human performance. Only Zhang et al. (2019) provide some quantitative analysis and report human performance. To

[1]The size of individual instances is important as it indicates the computational complexity of processing them. The longer the sequence the more time/memory it requires. This is important to keep in mind when applying methods with an input token limit, e.g. BERT (Devlin et al., 2019).

[2]I keep the terms the same way the authors use them but will explain more in Section 2.2.2

the best of my knowledge, none of the existing surveys looked into vocabulary and named entity analysis.

Methods and SOTA Description: A number of surveys (Liu et al., 2019a; Zhang et al., 2019; Wang, 2020; Qiu et al., 2019a; Baradaran et al., 2020; Thayaparan et al., 2020) include methods and report the SOTA results. Methods of reading comprehension are presented in Chapter 3. I considered adding SOTA for all datasets but decided against it because: (i) SOTA is changing quite fast, especially for the popular datasets; (ii) I wanted to focus on the datasets themselves. In Section 2.7 I look into SOTA and human performance for a subset of datasets to evaluate their difficulty.

	Zhang et al. (2019)	Liu et al. (2019a)	Qiu et al. (2019a)	Baradaran et al. (2020)	Wang (2020)	Dunietz et al. (2020)	Zeng et al. (2020)	Thayaparan et al. (2020)	This work (2020)
Datasets	23	15	11	29	13	27	47	17	**54**
Lang	EN&ZH	EN	EN	EN&ZH	EN&ZH	EN	EN	EN	EN
P clss	✗	✗	✗	✓	✓	✗	✓	✗	✓
Q clss	✗	✗	◆	✓	✗	◆	✓	✗	✓
A clss	✓	✓	✓	✓	✗	✓	✓	✗	✓
Domain	◆	✗	✗	◆	✗	✗	◆	◆	✓
HP	✗	✓	◆	✓	◆	◆	✓	✗	✓
Quant	✓	✗	✗	✗	✗	✗	◆	✗	✓
Vocab	✗	✗	✗	✗	✗	✗	✗	✗	✓
NE	✗	✗	✗	✗	✗	✗	✗	✗	✓
Methods	✓	✓	✓	✓	✓	✗	✗	✓	✗
Diffclt	✗	✗	✗	✗	✗	✓	✗	✗	✓
Struct	✓	✗	◆	✓	✗	✗	✓	◆	✓

Table 2.1: Comparison of this work with other MRC review papers including number of datasets (**Datasets**); language (EN and ZH stands for English and Chinese respectively); classification by passage (**P clss**), questions (**Q clss**), and answers (**A clss**); Availability of domain analysis (**Domain**), quantitative analysis (**Quant**), i.e. length of question/answer/passage; vocabulary (**Vocab**) and named entities (**NE**) analysis, description of the MRC methods (**Methods**); whether there is an analysis of datasets difficulty (**Diffclt**); and whether the information is presented in a structured way (**Struct**).
✓ – present; ◆ – present in a limited form; ✗ – not present.

Other: Thayaparan et al. (2020) focus the survey on *explainability* which is closely related to evaluation, generalisation, and interpretability. They consider 17 datasets discussing explainability itself, explanation-supporting benchmarks, and related architectures. Dunietz et al. (2020), considering 27 datasets, argue that existing understanding

of MRC is too unsystematic and propose a detailed definition of comprehension via *templates of understanding*.

2.2 Answer, Question, and Passage Type

All MRC datasets in this survey have three components: *passage*, *question*, and *answer*.[3] I begin with a categorisation based on the types of answers and the way the question is formulated. I divide questions into three main categories: *Statement*, *Query*, and *Question*. Answers are divided into the following categories: *Cloze*, *Multiple Choice*, *Boolean*, *Extractive*, *Generative*. The relationships between question and answer types are illustrated in Figure 2.2.

In what follows I briefly describe each question and answer category, followed by a discussion of passage types and dialog-based datasets.

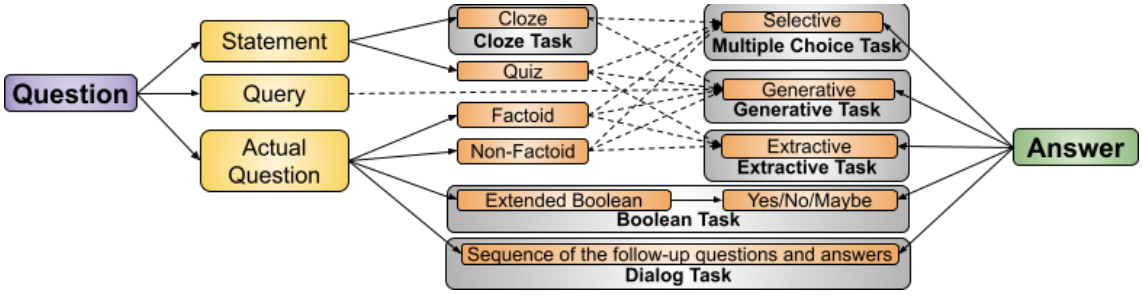


Figure 2.2: Hierarchy of question and answer types and the relationships between them. A solid arrow indicates a subtype whereas a dotted arrow indicates inclusion.

2.2.1 Answer Type

Cloze The question is formulated as a sentence with a missing word(s) and the correct entity should be inserted according to the context. The answer candidates may be included as in (2.1) from ReciteQA (Yagcioglu et al., 2018), and may not, as in (2.2) from CliCR (Šuster and Daelemans, 2018).

(2.1) **Passage:** *You will need 3/4 cup of blackberries ... Pour the mixture into cups and*

[3]There are a number of datasets that have been mentioned by previous surveys which do not meet this criteria. I mention them in the Appendix A.4 and explain why I did not include them in my analysis.

insert a popsicle stick in it or pour it in a posicle maker. Place the cup with the popsicle stick in it or the popsicle maker in the freezer. ...

Question: *Choose the best title for the missing blank to correctly complete the recipe. Ingredients, ___, Freeze, Enjoying*

Candidates: *(A) Cereal Milk Ice Cream (B) Ingredients (C) Pouring (D) Oven*

Answer: C

(2.2) **Passage:** *... However, intestinal perforation in dengue is very rare and has been reported only in eight patients until today. ...*

Question: *Perforation peritonitis is a ___.*

Possible answers: *very rare complication of dengue* **or** *very rare*

Multiple Choice (MC) A number of options are given for each question, and the correct one (or a number of correct answers) should be selected, e.g. (2.3) from MCTest (Richardson et al., 2013).

(2.3) **Passage:** *It was Jessie Bear's birthday. She was having a party. She asked her two best friends ...*

Question: *Who was having a birthday?*

Answer candidates: *(A) Jessie Bear (B) no one (C) Lion (D) Tiger*

Answer: A

I distinguish cloze multiple choice datasets from other multiple choice datasets. The difference is the form of the question: in the cloze datasets, the answer is a missing part of the question context and, combined together, they form a grammatically correct sentence, whereas for other multiple choice datasets, the question has no missing words.

Boolean A "Yes" or "No" answer is expected, e.g. (2.4) from the BoolQ dataset (Clark et al., 2019). Some datasets which I put in this category have the third option *"Cannot be answered"* or *"Maybe"*, e.g. (2.5) from PubMedQuestions (Jin et al., 2019).

(2.4) **Passage:** *The series is filmed partially in Prince Edward Island as well as locations in Southern Ontario (including Millbrook and Caledon).*

Question: *Is anne with an e filmed on pei?*

Answer: *Yes*

(2.5) **Passage:** *... Young adults whose families were abstainers in 2000 drank substantially less across quintiles in 2010 than offspring of non-abstaining families. The difference, however, was not statistically significant between quintiles of the conditional distribution. Actual drinking levels in drinking families were not at all or weakly associated with drinking in offspring. ...*

Question: *Does the familial transmission of drinking patterns persist into young adulthood?*

Answer: *Maybe*

Extractive or Span Extractive The answer is a substring of the passage. In other words, the task is to determine the start and end index of the characters in the original passage and the string between those two indexes is the answer, as shown in (2.6) from SQuAD (Rajpurkar et al., 2016).

(2.6) **Passage:** *With Rivera having been a linebacker with the Chicago Bears in Super Bowl XX, ...*

Question: *What team did Rivera play for in Super Bowl XX?*

Answer: *46-59: Chicago Bears*

Generative or Free Form Answer The answer must be generated based on information presented in the passage. Although the answer might be in the text, as illustrated in (2.7) from NarrativeQA (Kočiský et al., 2018), no passage index connections are provided.

(2.7) **Passage:** *...Mark decides to broadcast his final message as himself. They finally drive up to the crowd of protesting students, ... The police step in and arrest Mark and Nora....*

Question: *What are the students doing when Mark and Nora drive up?*

Answer: *Protesting*

2.2.2 Question Type

Statement is an affirmative sentence and is widely used in cloze tasks, e.g. (2.1)-(2.2). Quiz questions are also formulated in the form of a statement as in (2.8) from SearchQA (Dunn et al., 2017).

(2.8) **Passage:** *Jumbuck (noun) is an Australian English term for sheep, featured in Banjo Paterson's poem "Waltzing Matilda." Terminology. ...*
Question: *Australians call this animal a jumbuck or a monkey*
Answer: *Sheep*

Question is an actual question in the popular sense of the word, e.g. (2.3)-(2.7). Traditionally, questions are divided into Factoid (*Who? Where? What? When?*), Non-Factoid (*How? Why?*), and Yes/No.

Query is formulated to obtain a particular property of a particular object. It is similar to a knowledge graph query, and, in order to be answered, a part of the passage might involve additional sources, such as a knowledge graph, or the dataset may have been created using a knowledge graph, e.g. (2.9) from WikiReading (Hewlett et al., 2016).

(2.9) **Passage:** *Cecily Bulstrode (1584-4 August 1609), was a courtier and ... She was the daughter ...*
Question: *sex or gender*
Answer: *female*

2.2.3 Passage Type

Passages can take the form of a *one-document* or *multi-document* passage. They can be categorised based on the type of reasoning required in the passage to answer a question: *Simple Evidence* where the answer to a question is clearly presented in the passage, e.g. (2.3) and (2.6), *multihop reasoning* with questions requiring that several facts from different parts of the passage or different documents are combined to obtain the answer,

e.g. (2.10) from the HotpotQA (Yang et al., 2018b), and *extended reasoning* where general knowledge or common sense reasoning is required, e.g. (2.11) from the Cosmos dataset (Huang et al., 2019):

(2.10) **Passage:** *...2014 S\/S is the debut album of South Korean group WINNER. ... WINNER, is a South Korean boy group formed in 2013 by YG Entertainment and debuted in 2014. ...*

Question: *2014 S\/S is the debut album of a South Korean boy group that was formed by who?*

Answer: *YG Entertainment*

(2.11) **Passage:** *I was a little nervous about this today, but they felt fantastic. I think they'll be a very good pair of shoes. This time I'm going to keep track of the miles on them.*

Question: *Why did the writer feel nervous?*

Answer candidates:

(A) None of the above choices.

(B) Because the shoes felt fantastic.

(C) Because they were unsure if the shoes would be good quality.

(D) Because the writer thinks the shoes will be very good.

Answer: C

2.2.4 Conversational MRC

I include **Conversational** or **Dialog** datasets in a separate category as they involve a unique combination of passage, question, and answer. The full passage is presented as a conversation and the question should be answered based on previous utterances as illustrated in (2.12) from ShARC (Saeidi et al., 2018), where the scenario is an additional part of the passage unique for each dialog. The question and its answer become a part of the passage for the subsequent question.[4]

[4]I include DREAM (Sun et al., 2019) in the Multiple-Choice category rather than this category because, even though its passages are in dialog form, the questions are about the dialog but not a part of it.

(2.12) **Passage:** *Eligibility. You'll be able to claim the new State Pension if you're: a man born on or after 6 April 1951, a woman born on or after 6 April 1953*

Scenario: *I'm female and I was born in 1966*

Question: *Am I able to claim the new State Pension?*

Follow ups:

(1) *Are you a man born on or after 6 April 1951? – No*

(2) *Are you a woman born on or after 6 April 1953? – Yes*

Answer: *Yes*

2.2.5 Types of Datasets

Following the specified above classification, I provide seven categories of MRC QA datasets based on the task:

- **Cloze Datasets** are cloze types of questions with selective, or generative type answers;
- **Multiple-Choice Datasets** are statement-style, factoid and non-factoid questions with the selective answer type;
- **Generative Datasets** are query-style, statement-style, factoid and non-factoid questions with the generative answer type;
- **Extractive Datasets** are statement-style, factoid and non-factoid questions with the extractive answer type;
- **Boolean Datasets** are boolean questions with *Yes/No/Maybe* answers;
- **Dialog Datasets** are a sequential collection of questions and answers in various forms;
- **Mixed Datasets** are datasets with a mixture of questions from other task types.

Dataset	Size	Data Source	Q/A Source	LB	Human Performance	Solved	TMFW	PAD
Cloze Datasets								
CNN Hermann et al. (2015)	387k	CNN	AG	✻	-	✗	-	✓
Daily Mail Hermann et al. (2015)	997k	DailyMail	AG	✻	-	✗	-	✓
Children BookTest Hill et al. (2016)	687k	Project Gutenberg	AG	✻	82	✓	-	✓
Who Did What Onishi et al. (2016)	186k	Gigaword	AG	✓	84	✗	-	✉
BookTest Bajgar et al. (2017)	14M	Project Gutenberg	AG	✗	-	✗	-	✗
Quasar-S Dhingra et al. (2017b)	37k	Stack Overflow	AG	✗	46.8/50.0	✗	-	✓
RecipeQA Yagcioglu et al. (2018)	9.8k	instructibles.com	AG	✓	73.6	✗	-	✓
CliCR Šuster and Daelemans (2018)	105k	Clinical Reports	AG	✻	53.7/45.1(F1)	✗	-	✉
ReCoRD Zhang et al. (2018a)	121k	CNN	AG	✓✻	91.3/91.7(F1)	✓	-	✓
Shmoop Chaudhury et al. (2019)	7.2k	Project Gutenberg	ER, AG	✗	-	✗	-	✉
Multiple Choice Datasets								
MCTest Richardson et al. (2013)	2k/640	Stories	CRW	✓✻	95.3	✗	*what*	✓
WikiQA Yang et al. (2015)	3k	Wikipedia	UG, CRW	✻	-	✗	*what*	✓
bAbI Weston et al. (2016)	40k	AG	AG	✻	100	✓	*what*	✓

Continued on next page

Table 2.2 – *Continued from previous page*

Dataset	**Size**	**Data Source**	**Q/A Source**	**LB**	**Human Performance**	**Solved**	**TMFW**	**PAD**
MovieQA Tapaswi et al. (2016)	15k	Wikipedia	annotators	✓	-	✗	*what*	✉
RACE Lai et al. (2017)	98k	ER	experts	✓❇	73.3/94.5	✗	*what*	✓
SciQ Welbl et al. (2017)	12k	Science Books	CRW	✗	87.8	✗	*what*	✓
MultiRC Khashabi et al. (2018)	10k	reports, News, Wikipedia, ...	CRW	✓❇	81.8(F1)	✓	*what*	✓
MedQA Zhang et al. (2018b)	235k	Medical Books	expert	✗	-	✓	-	✗
MCScript Ostermann et al. (2018)	14k	Scripts, CRW	CRW	✗	98.0	✗	*how*	✓
MCScript2.0 Ostermann et al. (2019)	20k	Scripts, CRW	CRW	✗	97.0	✗	*what*	✓
RACE-C Liang et al. (2019)	14k	ER	experts	✗	-	✗	*the*	✓
DREAMSun et al. (2019)	10k	ER	experts	✓	98.6	✗	*what*	✓
Cosmos QA Huang et al. (2019)	36k	Blogs	CRW	✓	94	✗	*what*	✓
ReClor Yu et al. (2020)	6k	ER	experts	✓	63.0	✗	*which*	✓

Continued on next page

Table 2.2 – *Continued from previous page*

Dataset	Size	Data Source	Q/A Source	LB	Human Performance	Solved	TMFW	PAD
QuAIL Rogers et al. (2020a)	15k	News, Stories, Fiction, Blogs, UG	CRW, experts	✓	60.0	✗	-	✓
Boolean Questions								
BoolQ Clark et al. (2019)	16k	Wikipedia	UG, CRW	✓✻	89	✓	*is*	✓
AmazonYesNo Dzendzik et al. (2019)	80k	Reviews	UG	✗	-	✗	*does*	✉
PubMedQA Jin et al. (2019)	211k	PubMed	CRW	✓	78	✗	*does*	✓
Extractive Datasets								
SQuAD Rajpurkar et al. (2016)	108k	Wikipedia	CRW	✓✻	86.8(F1)	✓	*what*	✓
SQuAD2.0 Rajpurkar et al. (2018)	151k	Wikipedia	CRW	✓✻	89.5(F1)	✓	*what*	✓
NewsQA Trischler et al. (2017)	120k	CNN	CRW	✻	69.4(F1)	✓	*what*	✓
SearchQA Dunn et al. (2017)	140k	CRW, AG	J!Archive	✻	57.6(F1)	✓	*this*	✓
Generative Datasets								
MS MARCO Nguyen et al. (2016)	100k	Web documents	UG, HG	✓✻	-	✗	*what*	✓

Continued on next page

Table 2.2 – *Continued from previous page*

Dataset	Size	Data Source	Q/A Source	LB	Human Performance	Solved	TMFW	PAD
LAMBADA Paperno et al. (2016)	10k	BookCorpus	CRW, AG	✗	-	✗	-	✓
WikiMovies Miller et al. (2016)	116k	Wikipedia, KG	CRW, AG, KG	✗	93.9 (hit@1)	✗	*what*	✓
WikiSuggest Choi et al. (2017)	3.47M	Wikipedia	CRW, AG	✗	-	✗	-	✗
TriviaQA Joshi et al. (2017)	96k	Wikipedia, Web docs	Trivia, CRW	✓✻	79.7/75.4 wiki/web	✗	*which*	✓
NarrativeQA Kočiský et al. (2018)	47k	Wikipedia, movie, HG, Project Gutenberg	HG	✻	19.7 BLEU4	✓	*what*	✓
TweetQA Xiong et al. (2019)	14k	News, Twitter, HG	CRW	✓	70.0 BLEU1	✓	*what*	✓
Conversational Datasets								
ShARC Saeidi et al. (2018)	32k	Legal web sites	CRW	✓	93.9	✗	*can*	✓
CoQA Reddy et al. (2019)	127k	Books, News, Wikipedia, ER	CRW	✓✻	88.8	✓	*what*	✓
Mixed Datasets								

Continued on next page

Table 2.2 – *Continued from previous page*

Dataset	Size	Data Source	Q/A Source	LB	Human Performance	Solved	TMFW	PAD
TurkQA Malon and Bai (2013)	54k	Wikipedia	CRW	✗	-	✗	*what*	✓
WikiReading Hewlett et al. (2016)	18.9M	Wikipedia	AG, KG	✗	-	✗	-	✓
Quasar-T Dhingra et al. (2017b)	43k	Trivia ClueWeb09	AG	✻	60.4/60.6(F1)	✗	*what*	✓
HotpotQA Yang et al. (2018b)	113k	Wikipedia	CRW	✓✻	96.37(F1)	✗	*what*	✓
QAngaroo WikiHop Welbl et al. (2018)	51k	Wikipedia	CRW, KG	✓✻	85.0	✗	-	✓
QAngaroo MedHop Welbl et al. (2018)	2.5k	Medline abstracts	CRW, KG	✓	-	✗	-	✓
QuAC Choi et al. (2018)	98k	Wikipedia	CRW	✓✻	81.1(F1)	✗	*what*	✓
DuoRC Saha et al. (2018)	86k	Wikipedia + IMDB	CRW	✓	-	✗	*who*	✓
emr QA Pampari et al. (2018)	456k	Clinic Records	expert, AG	✗	-	✗	*does*	✉
DROP Dua et al. (2019)	97k	Wikipedia	CRW	✓	96.4(F1)	✗	*how*	✓
NaturalQuestions Kwiatkowski et al. (2019)	323k	Wikipedia	UG, CRW	✓✻	87/76 L/S(F1)	✗	*who*	✓

Continued on next page

Table 2.2 – *Continued from previous page*

Dataset	**Size**	**Data Source**	**Q/A Source**	**LB**	**Human Performance**	**Solved**	**TMFW**	**PAD**
AmazonQA Gupta et al. (2019b)	570k	UG Review	UG	✗	53.5	✗	*does*	✓
TyDi Clark et al. (2020)	11k	Wikipedia	CRW	✓	54.4(F1)	✗	*what*	✓
R^3 Wang et al. (2020b)	60k	Wikipedia	CRW	✗	-	✗	-	✗

Table 2.2: Reading comprehension datasets comparison. Where (abbreviations are listed in order of appearance in the table): **LB** – leader board available; **Human Performance** (expert/non-expert if other not specified): accuracy if other is not specified; **TMFW** – the most frequent first word; **PAD** – publicly available data; **k/M** – thousands/millions; **CRW** – crowdsourcing; **AG** – automatically generated; **KG** – knowledge graph; **ER** – educational resources; **UG** – user generated; **HG** – human generated (UG + annotators, crw, experts); **L/S** – long/short answer; ✓ – available/"solved"; ✗ – unavailable/not "solved"; ❇ – the leader board is presented at `https://paperswithcode.com/`; ✉ – the dataset is available by request. The information is verified in June 2021.

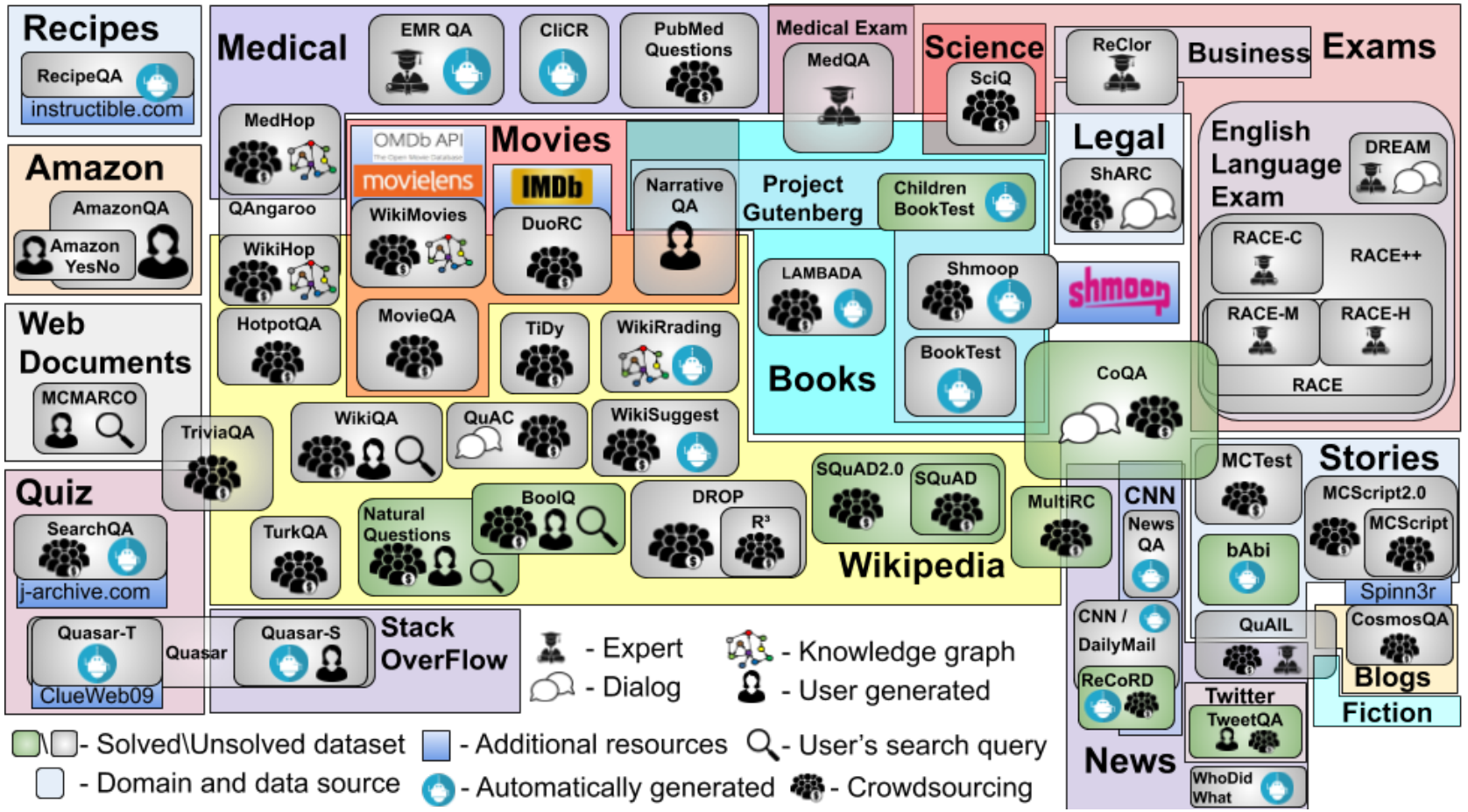


Figure 2.3: Question Answering Reading Comprehension datasets overview.

2.3 Datasets

All datasets categorized by tasks and their properties of interest are presented in Table 2.2. I indicate the number of questions per dataset (size), the text sources, the method of creation, whether there is a leaderboard and data publicly available, and whether the dataset is *solved*, i.e. the performance of an automatic RC system exceeds the reported human performance (also shown). I will discuss each of these aspects in turn in the following sections.

2.3.1 Data Sources

A significant proportion of datasets (21 out of 54) use Wikipedia as a passage source. Six of those use Wikipedia along with additional sources. Other popular sources of text data are:

1. news (CNN/DailyMail, WhoDidWhat, NewsQA, CoQA, MultiRC, ReCoRD, and QuAIL);
2. books, including Project Gutenberg[5] and BookCorpus[6] (Zhu et al., 2015) (Children-BookTest, BookTest, LAMBADA, partly CoQA, Shmoop, SciQ);
3. movie scripts (MovieQA, WikiMovies, DuoRC), and in combination with books (MultiRC and NarrativeQA);
4. medicine: five datasets (CliCR, PubMedQuestions, MedQA, emrQA, QAngaroo MedHop) were created in the medical domain based on clinical reports, medical books, MEDLINE abstracts and PubMed;
5. exams: RACE, RACE-C, and DREAM use data from English as a Foreign Language examinations, ReClor from the Graduate Management Admission Test (GMAT)[7] and The Law School Admission Test (LSAT), [8] and MedQA from medical exams.

[5] `www.gutenberg.org` – last verified May 2021
[6] `yknzhu.wixsite.com/mbweb` – last verified May 2021
[7] `www.mba.com/exams/gmat/` – last verified May 2021
[8] `www.lsac.org/lsat` – last verified May 2021

Other sources of data include legal resources websites[9] (ShARL), personal narratives from the Spinn3r Blog Dataset (Burton et al., 2009) (MCScript, MCScript2.0, CosmosQA), StackOverflow.com (Quasar-S), Quora.com (QuAIL), Twitter.com (TweetQA),[10] Amazon.com user reviews and questions (AmazonQA, AmazonYesNo), and a cooking website[11] (RecipeQA).

Figure 2.3 shows the domains used by datasets as well as any overlaps between datasets. Some datasets share not only text sources but also the actual questions themselves. SQuAD2.0 was created as an extension of SQuAD with unanswerable questions. AmazonQA and AmazonYesNo share the same set of questions and reviews with different processing.[12] BoolQ shares 3k questions and passages with the NaturalQuestions dataset. The R^3 dataset is fully based on DROP with a focus on reasoning.

2.3.2 Dataset Creation

Rule-based approaches have been used to automatically obtain questions and passages for the MRC task by generating the sentences (e.g. bAbI) or, in the case of cloze type questions, excluding a word from the context. I call those methods *automatically generated* (AG). Most dataset creation, however, involves a human in the loop. I distinguish three types of people: *experts* are professionals in a specific domain; *crowdworkers* (CRW) are casual workers who normally meet certain criteria (for example a particular level of proficiency in the dataset language) but are not experts in the subject area; *users* who voluntarily create content based on their personal needs and interests.

More than half of the datasets (31 out of 54) were created using crowdworkers. In one scenario, crowdworkers have access to the passage and must formulate questions based on it. For example, MovieQA, ShaRC, SQuAD and SQuAD2.0 were created in this way. In contrast, another scenario involves finding a passage containing the answer to a given question. That works well for datasets where questions are taken from already

[9]For example: `www.benefits.gov/`, `www.gov.uk/`, `www.usa.gov/` – all links last verified May 2021

[10]Xiong et al. (2019) selected tweets featured in the news.

[11]`www.instructables.com/cooking` – last verified May 2021

[12]I explain the difference in details in Chapter 7.

existing resources such as trivia and quiz questions: TriviaQA, Quasar-T, and SearchQA, or using web search queries and results from Google and Bing as a source of questions and passages: BoolQ, NaturalQuestions, MSMARCO. In the ReCoRD dataset, the questions, answers, and passages were created automatically but crowdworkers were used to filter ambiguous and noisy samples.

In an attempt to avoid repetition of words and phrases between passages and questions, some datasets used different texts about the same topic as a passage and a source of questions. For example, DuoRC takes descriptions of the same movie from Wikipedia and IMDB.[13] One description is used as a passage while another is used for creating the questions, e.g. (2.13):

(2.13) **Movie:** *Kung Fu Panda (2008)*

> **Context** (from IMDB):[14] *... Using ingenuity and his rebounding stomach, Po manages to render Tai Lung helpless. Tai Lung spews out a slur of insults before Po takes his finger in the Wuxi Finger Hold. ...*
>
> **Wikipedia article**:[15] *... Eventually, Po defeats Tai Lung in combat by using the mysterious Wuxi Finger Hold to vanquish him. ...*
>
> **Question:** *What move does Po use to finish off Tai Lung?*
>
> **Answer:** *Wuxi Finger Hold*

NewsQA uses only a title and a short news article summary as a source of questions while the whole text becomes the passage. Similarly, in NarrativeQA, only the abstracts of the story were used for question creation. For MCScript and MCScript 2.0, questions and passages were created by different sets of crowdworkers given the same script from script data collections (Modi et al., 2016; Regneri et al., 2010; Singh et al., 2002; Wanzare et al., 2016; Burton et al., 2009).

[13] `https://www.imdb.com/` – last verified June 2021

[14] `https://www.imdb.com/title/tt0441773/plotsummary?ref_=tt_stry_pl#synopsis` – last verified June 2021

[15] `https://en.wikipedia.org/wiki/Kung_Fu_Panda` – last verified April 2020

2.3.3 Additional Features

Table A.2[16] in the Appendix contains additional features of datasets such as:

- whether the datasets questions are boolean, non-factoid, without answer, or in the query form;
- whether multi-hop reasoning is required to answer the question;
- whether it is a multi-document dataset or dialogues are the part of the passage;
- whether there is any additional data available within the dataset, e.g. if it is multilingual or multimodal.

Figure 2.4 reflects a summary of additional features present in the datasets. A feature that is present in a limited form (yellow part of the chart) means there are some samples which happen to have this property but it was not strongly intentional. Just about half of all datasets focus on non-factoid question and slightly less than half focus on questions where more than one sentence is needed to answer the question (Multi Hop). Another nine datasets have some questions where multi hop reasoning is needed, although most samples from those datasets can be resolved based on one sentence. Apart from those three datasets, which are precisely listed as boolean, there are a significant amount of boolean questions in 14 other datasets, including ShARC which is a conversational dataset where all question are boolean. Also seven more datasets contain some boolean questions. 14 datasets were designed with unanswerable questions. For example, the availability of samples without an answer is the main difference between the SQuAD and SQuAD2.0 datasets. Apart from those two datasets which are precisely listed as conversational, three more datasets contains dialogues. Four datasets use questions in a query format. Finally, seven datasets are multimodal or multilingual.

[16]While Table 2.2 contains datasets grouped by task and ordered by year of appearance, to keep track of all datasets in Table A.2 and future tables I list them in alphabetical order, excluding Shmoop and WikiSuggest as those two datasets are not publicly available, but including BookTest, MedQA, and R^3 as all required information is available in the original papers.

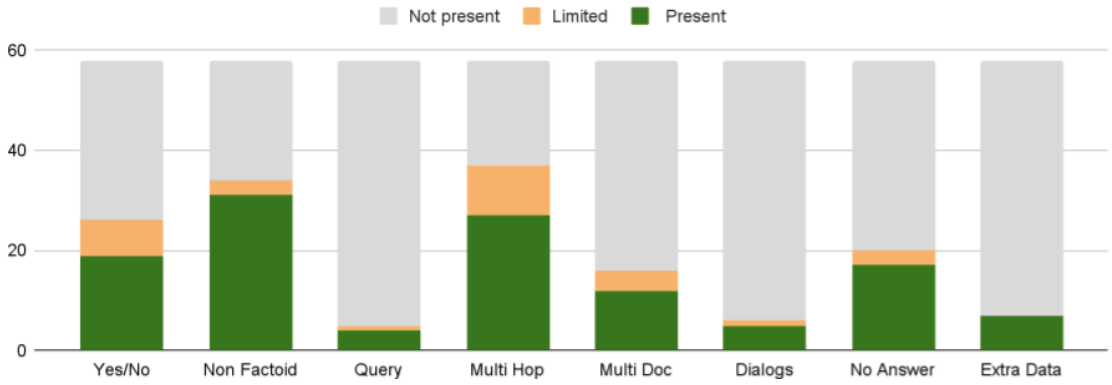


Figure 2.4: Overview of additional dataset properties.

2.4 Quantitative Analysis

Each dataset's size is shown in Table 2.2. The majority of datasets contain 100k+ questions which makes them suitable for training and/or fine tuning a deep learning model. A few datasets contain fewer than 10k samples: MultiRC (9.9k), Shmoop (7.2k), ReClor (6.1k), QAngaroo MedHop (2.5k), WikiQA (2k). Every dataset has its own structure and data format but I processed all datasets the same way, extracting lists of questions, passages, and answers including answer candidates, and using the *Stanza*[17] library, which is developed by Qi et al. (2020). Stanza was selected over the *spaCy*[18] tokenizer as it outperform it in both token and sentence tokenization on the Universal Dependencies (v2.5) English treebank[19], showing promising F1 scores of 99.01 (tokens) and 81.13 (sentences) (Qi et al., 2020).

The graphs in Figure 2.5[20] provide more insight into the differences between the datasets in terms of answer, question and passage length, as well as vocabulary size.[21]

These statistics represent an important characteristic of the dataset itself. As was mentioned above, they contribute to the dataset difficulty. The aggregation of passage, question and answer lengths shows the common trends in datasets in general, and highlights those which are different from the majority which helps to answer the first research

[17] `https://stanfordnlp.github.io/stanza/` – last verified November 2021
[18] `spacy.io/api/tokenizer` – last verified November 2021
[19] `https://universaldependencies.org/en/index.html` – last verified November 2021
[20] I use `matplotlib` for calculation and visualisation: `https://matplotlib.org/` – last verified June 2021
[21] See Table A.2 in Appendix for more details

question.

The majority of datasets have a passage length under 1500 tokens with the median being 329 tokens but due to seven outliers the average number tokens is 1250 (Figure 2.5 (a)). Some datasets (MS MARCO, SearchQA, AmazonYesNo, AmazonQA, MedQA) have a collection of documents as a passage but others contain just a few sentences. The number of tokens in a question lies mostly between 5 and 20. Two datasets, ChildrenBookTest and WhoDidWhat, have on average more than 30 tokens per question while WikiReading, QAngaroo MedHop and WikiHope have only 2 – 3.5 average tokens per question (Figure 2.5 (b)). The majority of datasets contain fewer than 8 tokens per answer with the average being 3.5 tokens per answer. The outlier is NaturalQuestions which has on average 164 tokens per answer[22] (Figure 2.5 (c)).

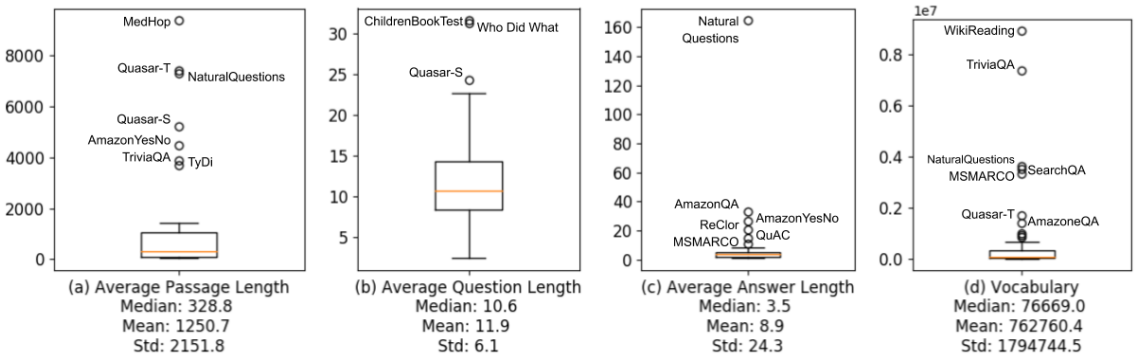


Figure 2.5: The average length in tokens of (a) passages, (b) questions, (c) answers, and (d) vocabulary size in unique lower-cased lemmas of datasets with the median (**Median**), mean value (**Mean**), and standard deviation (**Std**). Outliers are highlighted.

2.4.1 Types of Questions and Question Words

A number of datasets come with a breakdown of question types based on the first token (Nguyen et al., 2016; Ostermann et al., 2018, 2019; Kočiský et al., 2018; Clark et al., 2019; Gupta et al., 2019b). I inspected the most frequent first word in a dataset's questions excluding cloze-style questions. Table 2.2 shows the most frequent first word per dataset and Table A.3 shows the same information over all datasets. The most popular first word is *what* – 22% of all questions analysed and over half of questions in WikiQA, Wiki-

[22]I focus on short answer, considering a long one only if the short answer is not available.

Movies, MCTest, CosmosQA, and DREAM start with *what*. The majority of questions in ReClor (56.5%) start with the word *which*, and RACE has 23.1%. DROP mostly focused on *how much/many, how old* questions (60.4%). DuoRC has a significant proportion of *who/whose* questions (39.5%). *Why*, *When*, and *Where* questions are under-represented – only 1.4%, 2%, and 2.3% of all questions respectively. Only CosmosQA has a significant proportion (34.2%) of *Why* questions, MCScript2 (27.9%) and TyDi (20.5%) of *When* questions, and bAbI (36.9%) of *Where* questions.

Some questions can be formulated with the question word contained within, for example: *"About how much does each box of folders weigh?"* or *"According to the narrator, what may be true about their employer"*. I automatically analyse 6.7M questions excluding all cloze datasets (there are all together approximately 2.5M cloze questions) and WikiReading, WikiHop, and MedHop (almost 19 million questions-queries), as the queries are not formulated in question form. As mentioned in section 2.3.1 some datasets shared the questions and some datasets have the same questions asked more than once within a different context (for example, question *"Where is Daniel?"* asked 2007 times in bAbI), or the same questions asked with different answer options (for example in CosmosQA dataset). I calculated the frequency of question words for both scenarios: *all questions* and *unique questions* (see Table 2.3).

Apart from datasets which contain only Yes/No/Maybe questions[23] a significant portion of boolean questions are in ShaRC (85.4%), emrQA (74.0%) AmazonQA (55.3%), QuAC (36.6%), MCSCript (28.6%), TurcQA (25.7%), bAbI (25.0%) and CoQA (20.7%). Almost a third of all questions and more than quarter of unique questions are boolean. Another quarter of unique questions (26.57%) contain the word *What*, 6.64% of questions asks *who* and *whose*, and 4.49% *which*, about 3% of questions are *when* and *where*. Only 5.95% ask the question *how* excluding (*how many/much* and *how old*). Other questions which do not contain any of these question words constitute 16.73% of unique questions.[24]

[23]To separate boolean questions I used the same list of words as Clark et al. (2019): *did, do, does, is, are, was, were, have, has, can, could, will, would*.

[24]There are datasets where more than 20% of questions are formulated in such a way that the first token is not one of the considered words: Quasar-S (98.8%), SearchQA (98.3%), RACE-C (64.1%), TriviaQA (49.6%), HotPotQA (42.0%), Quasar-T (40.7%), MSMARCO (26.6%), NaturalQuestions(23.4%), Ama-

None of the analyzed data sets has a balanced distribution of considered question words. See Table A.3 in Appendix for more detailed information.

Question	All Questions				Unique Questions			
	First token		Contains		First token		Contains	
	Count	%	Count	%	Count	%	Count	%
what	1497009	22.39	1687898	25.25	1069275	24.23	1172454	26.57
when	137865	2.06	156628	2.34	116158	2.63	131509	2.98
where	154990	2.32	166866	2.50	119250	2.70	128067	2.90
which	275454	4.12	541835	8.10	123731	2.80	198335	4.49
why	95493	1.43	98649	1.48	68217	1.55	71185	1.61
how	258961	3.87	298526	4.47	230948	5.23	262646	5.95
who/whose	392166	5.87	392166	5.87	293130	6.64	293130	6.64
how much/ many/ old	197598	2.96	197598	2.96	157427	3.57	157427	3.57
boolean	2236356	33.45	-	-	1259287	28.53	-	-
other	1439241	21.53	907748	13.58	975681	22.11	738245	16.73

Table 2.3: Frequency of first token of the questions and question words inside the question across datasets.

2.4.2 Vocabulary

To obtain a vocabulary size I calculate the number of unique lower cased lemmas of tokens. A vocabulary size distribution is presented in Figure 2.5 (d). There is a moderate correlation[25] between the number of questions in a dataset and its vocabulary size (see Figure 2.6). WikiReading has the largest number of questions as well as the richest vocabulary. bAbI is a synthetic dataset with 40k questions but a vocabulary of only 152 lemmas for both question and passage.

Observing a huge difference in vocabulary sizes (the y-axis is logarithmic in Figure 2.6), I decided to explore the vocabulary further. I ran a language detector over all datasets using the python `pyenchant` library[26] for American (US) and British English,

zonQA (22.8%), and SQuAD (21.1%).

[25]As the data has a non-normal distribution I applied the Spearman correlation coefficient $= 0.58$ and $p - value = 1.3e - 05$. The guide for interpreting the Spearman correlation coefficient, according to `https://www.statstutor.ac.uk/resources/uploaded/spearmans.pdf` – last verified October 2021, is 0.00-0.19 "very weak", 0.20-0.39 "weak", 0.40-0.59 "moderate", 0.60-0.79 "strong", 0.80-1.0 "very strong". The correlation is computed with `scipy.stats` python package: `https://docs.scipy.org/doc/scipy/reference/reference/stats.html` – last verified October 2021

[26]`https://pypi.org/project/pyenchant/` – last verified June 2021.

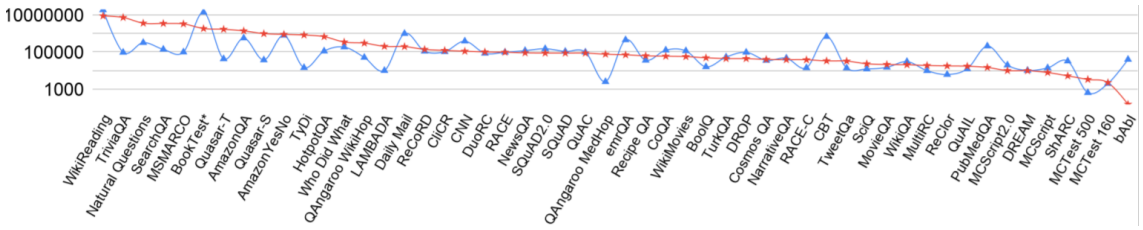


Figure 2.6: The number of questions (blue triangle ▲) and vocabulary size (unique lowercased lemmas) (red star ★). The values for the BookTest (Bajgar et al., 2017) and WhoDidWhat (Onishi et al., 2016) are borrowed from correspondent papers.

and `langid` library.[27]

I divided all lemmas in the vocabulary into five categories:

- *English Words* are regular English words excluding names and other named entities but including time (*10:00pm, gmt+06:42:04*) and other measurements (*-220v*);
- *Non-English Words* – this category contains foreign words transliterated into English, named entities, slang (*aaaaaaand*, *aaaarrrggh*), and spelling mistakes (*dateing*, *fantstic*). It also contains a mixture of common ASCII symbols and/or letters. Words such as colour code (*color:#ffffa0*), and other (*forest_biome[1][1].ppt*) are in this category. Those words normally do not make any sense without context;
- *Numbers* which also can contain symbols: . , + - $ £% and € indicating regular numbers, prices, and percentage, for example $11, 788, 232$, 11$, +3, etc.;
- *Non-ASCII* is a category of tokens which contains symbols outside of usual ASCII table (ASCII code >128). This category contains mostly foreign words in their original language, special symbols, and emoji.
- Finally, I separated *Web links* into their own category.[28]

Undoubtedly, the first two categories of lemmas (English words and Numbers) are essential for understanding and answering the questions as they encapsulate the main context and values. The last two categories (Non-ASCII and Web links) can be categorised as additional data or noise as normally this information is not needed for general

[27] `https://github.com/saffsd/langid.py` – last verified June 2021.

[28] I used a primitive check with regular expressions for separating the web links prioritizing Precision over Recall.

understanding unless there is a very specific question asked (as an example (2.14) from Natural Questions) but such questions are rare. Table A.1 in Appendix shows more detailed information per dataset.

(2.14) **Question:** *what do the 3 dots mean in math?*

Simplified Context[29]: *In logical argument and mathematical proof, the therefore sign* (∴) *is generally used before a logical consequence, such as the conclusion...*

In 36 of the 54 datasets, more than 10% of the words are reported to be non-English. This category also contains lemmas which I call ASCII mix which have both letters from English alphabet and other ASCII symbols. The high amount of ASCII mix lemmas can also indicate a big percentage of tokenization and spelling errors as it contains such tokens as ("*yummy!if*", "~~~", "*you!!!seriously*", etc.) in RecipeQA, AmazonQA, AmazonYesNo. This is a more common case for user generated data rather then crowdsourced. Also, there are a number of words in the vocabulary which are transliterated from different languages.[30]

I inspected 200 samples each (chosen at random) from a subset of these.[31] For Wikipedia datasets (HotPotQA, QAngaroo WikiHop), around 70-75% of those words are named entities; 10-12% are specific terms borrowed from other languages such as names

[29]Original context includes HTML tags so I removed it to make example better understandable.

[30]For example, word *zvezda* (Russian: звезда – noun, means star) is presented in 11 datasets (DailyMail, DuoRC, HotpotQA, MSMarco, NaturalQuestions, Quasar-T, SearchQA, TriviaQA, TyDi, WikiHop, and WikiReading). Almost half of those datasets also contain a diminutive form of the word: *zvezdochka* (Russian: звездочка – noun, diminutive form, means little star). Unlike English, where nouns only have two grammatical cases: nominative and genitive, the Russian language has six. Four of those datasets which are fully or partly based on Wikipedia (WikiReading, NaturalQuestions, TriviaQA, and TyDi) also contain different grammatical forms of this word as *zvezdam* (Russian: звездам – dative case, plural), *zvezdami* (Russian noun: звездами – instrumental case, plural), *zvezdah* (Russian: звездах – prepositional case, plural). Here and in future examples I do not exclude a possibility that those words could have a different meaning and be different grammatical form in other languages. Examples of other transliterated words were spotted in previously mentioned datasets such as *changcheng* (Chinese: 长城 , pinyin (the Romanization of the Chinese characters based on their pronunciation): chang cheng – the great wall); *guandi* (Chinese: 官邸 , pinyin: guan di – mansion); *sicherheitsdienstleistungen* (German noun, means security service) spotted only in NaturalQuestions and TriviaQA; *dignidad* (Spanish noun: means dignity); *descanso* (Spanish noun: means break); *verano* (Spanish noun: means summer) spotted in all datasets mentioned above as well as in AmazonQA, LAMBADA, DailyMail, SQuAD and SQuAD2.0.

[31]I used the following resources (all links last verified May 2021): Google search (`https://google.com`), Google Translate (translate.google.com), Wikipedia (`https://www.wikipedia.org`), WikiDictionary (`https://www.wiktionary.org`), Collins Dictionary (`https://www.collinsdictionary.com`), The Free Dictionary (`https://www.thefreedictionary.com`), Amazon (`https://www.amazon.com`)

of plants animals, etc.; another 8-10% are foreign words, e.g. the word *dialetto* from HotPotQA *"Bari dialect (dialetto barese) is a dialect of Neapolitan ..."*; about 1.5-3% are misspelled words and tokenization errors. In contrast, for the user-generated dataset, AmazonQA, 67% are tokenization and spelling errors.

2.5 Named Entities

Named entities (NE) are an important part of reading comprehension as they often define the information which is looked for when the question is asked. They provide a well defined categories of words such as names, locations, dates, quantity, etc. Some well known NEs might imply some particular general knowledge about this entity. Additional information about connected named entities can be found in knowledge graphs and be used for answering the question. I use the following NE categorisation scheme:

- *Person* – name of a person;
- *Organisation* – name of organisation or company;
- *GPE LOC* – location or geographical object;
- *Extra* – additional named entities such as products, works of art, law, language, etc;
- *Date/Time (+% $)* – date, time, percentage, and money related named entities;
- *Misc* – other numerical named entities such as quantity, ordinal, and cardinal.

2.5.1 Common Named Entities

I extracted the named entities with the *stanza* library. Table A.4 (left and middle parts) contains detailed information per dataset including:

- the total number of named entities per dataset;
- the number of unique named entities per dataset;

- the average number of named entities per question and passage;
- the percentage of samples where there is no named entities either in the question or passage;
- the percentage of samples where the same named entity is mentioned in both the question and passage

There are a few datasets with more than 10 million named entities mentioned: DailyMail, HotPotQA, AmazonQA, and TriviaQA. The majority of datasets have fewer than one named entity per question. Questions with on average more than one named entity per question are asked in the following datasets: CNN, DailyMail, DROP, MovieQA, MultiRC, NarrativeQA, ReCoRD, both SQuADs, and TurkQA. Datasets with a higher number of named entities per question are HotPotQA (2.4), TriviaQA (2.5) and WhoDidWhat (5).

Table A.4 (right part) shows the breakdown of named entities by type. As expected, for some domain specific datasets (products: AmazonQA and AmazonYesNo; medical: Qangaroo MedHop and PubMedQuestions; cooking: RecipeQA; legal: ShaRC; science: SciQ) and blogs (MCScript and MCScript2.0) the percentage of common named entities (person, organisation, and location) is under 30%. The medical and science datasets also have a high proportion (over 30%) of other numerical (*Misc*) NE such as weight or distance measures, ordinal, etc.

2.5.2 Named Entities in Questions and Answers

Table A.5 shows a detailed breakdown of NE types in questions across datasets. Figure 2.7 (a) shows the average percentage of questions with a particular NE per domain.[32]

All domains except `Product` have more than 10% of questions which contain the *Person* named entity. At least 30% of questions in the `News`, `Story`, and `Wikipedia` domain (including a combination of Wikipedia with other resources) the *Person* NE. As

[32]I exclude medical domain from this analysis for two reasons: (1) It is computationally expensive to identify named entities in medical datasets, and (2) medical named entity itself are very different from other and requires deeper analysis which is beyond the scope this thesis. In the meantime, Zhang et al. (2021) presented a new version of Stanza for biomedical and clinical data which could be used for future analysis.

expected, Products has a larger number (over 20%) of questions from Misc named entities. News generally has over 15% of questions with all types of named entities, while for other domains more than half of named entity types are present in less than 10% of questions, especially the Story domain where the percentage of questions with named entities other then *Person* is under 5%.

I repeat the same analysis for answers. Table A.6 in Appendix shows a detailed breakdown of NE types in answers across datasets. Figure 2.7 (b) reflects the average percentage of answers with certain named entities per domain. The *Person* NE is found, on average, in over 50% of answers in News domain, which is expected, while in Wikipedia datasets around 20% or more of answers also contain a *Person* type named entity. Products, also as expected, has a large proportion of answers with *Misc* NE.

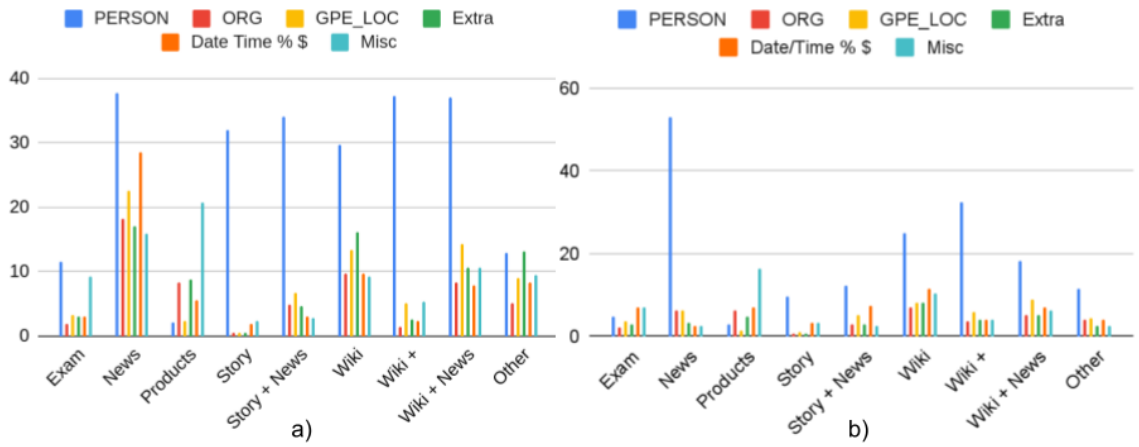


Figure 2.7: Average percentage of (a) questions which contain a particular NE, and (b) answers which contain a particular NE

2.6 Evaluation and Human Performance

In this section I present evaluation metrics used to evaluate human and machine reading comprehension.

2.6.1 Evaluation Metrics

In Table 2.2, I report Human Performance (HP) using accuracy as an evaluation metric. If other metrics are used to evaluate systems trained on a dataset, it is specified in the brack-

ets. *Accuracy* is defined as the ratio of correctly answered questions out of all number of questions as Equation (2.15) shows:

$$Accuracy = \frac{|Q_{correct}|}{|Q|} \tag{2.15}$$

where $Q_{correct}$ is a set of correctly answered questions and Q is a set of all questions.

Let's consider an example of Wikipedia-based passage (2.16) and a number of questions (2.16-2.21). For those datasets where the answer should be selected, accuracy is the natural and the most common metric. If there is a system which answers the questions (2.18) and (2.19); the performance of the system can be calculated using Equation (2.15).

(2.16) **Passage**:[33] *The giant panda, also known as the panda bear or simply the panda, is a bear native to south central China. Though it belongs to the order Carnivora, the giant panda is a folivore, with bamboo shoots and leaves making up more than 99% of its diet. Although scientists do not know why these unusual bears are black and white, speculation suggests that the bold colouring provides effective camouflage in their shade-dappled snowy and rocky habitat. They are able to climb and take shelter in hollow trees or rock crevices, but do not establish permanent dens. For this reason, pandas do not hibernate, which is similar to other subtropical mammals, and will instead move to elevations with warmer temperatures. In March 2015, conservation news site Mongabay stated that the wild giant panda population had increased by 268, or 16.8%, to 1,864.*

(2.17) **Question:**[34] *Pandas mainly eat* _____

Answer (cloze): *Bamboo*

(2.18) **Question:** *Why are pandas black and white?*

Answer candidates: *(A) Camouflage and communication (B) Courting (C) Heat management (D) To fend off predators*

Answer (selective): A

[33]The passage is created based on Giant panda Wikipedia Page `https://en.wikipedia.org/wiki/Giant_panda` – last verified January 2021

[34]Source of questions `https://www.wwf.org.uk/node/40546` – last verified January 2021

(2.19) **Question:** *Do they hibernate?*

Answer (boolean): No

(2.20) **Question:** *Where are wild giant pandas found?*

Answer (extractive): *100-104 (China)*

(2.21) **Question:** *How many giant pandas remain in the wild?*

Answer (generative): *1860*[35]

For those datasets where the answer should be found or generated (extractive or generated tasks) accuracy is the same as *Exact Match* (EM) implying the answer provided by a system is exactly the same as the gold answer.

In contrast with selective and boolean tasks, extractive or generative tasks can have ambiguous, incomplete, or redundant answer, e.g. for question (2.20) the system might provide more a detailed answer such as *south central China*. Or, based on passage (2.16), the answer *1864* to question (2.21) might be generated. In order to assign credit when the system answer does not exactly match the gold answer, the *F-1 measure* can be used.

For cloze datasets the metrics depends on the form of answer. If there are options available the accuracy can be calculated. If there are missed words that have to be generated the exact match and F-1 measure can be applied.

F-1 measure is a harmonic mean of *Precision* (P) and *Recall* (R). See (2.22). It can be calculated on the level of words:

$$P = \frac{|TP|}{|TP| + |FP|}, R = \frac{|TP|}{|TP| + |FN|} \tag{2.22}$$

where *TP (True Positive)* are the words from gold answer matching the predicted answer; *FP (False Positive)* are the words from predicted answer which are not in the gold answer; and *FN (False Negative)* are the words from gold answer which are not in the predicted answer. Additionally, like in Example (2.21), F1-measure can be calculated on a character level.

[35] As the question-answer pair and the passage are from different sources, the information does not match exactly.

Therefore, to evaluate those cases where the predicted answer is partly correct the *macro-averaged F-1* measure is used as an average F-1 measure of predicted tokens over all questions. Example of accuracy (exact match) and F1-measure evaluation is provided in Table 2.4.

Question	**Gold Answer**	**Example of Predicted Answer**	**EM**	**F-1**
Do they hibernate?	Yes	Yes	1	1
Where are wild giant pandas found?	China	south central China	0	0.5
Why are pandas black and white?	Camouflage, communication	Camouflage	0	0.67
How many giant pandas remain in the wild?	1860	4242	0	0
Overall			0.25	0.5425

Table 2.4: Example of EM and F-1 calculation

Accuracy is used for all boolean, multiple choice,[36] and cloze datasets except CliCR and ReCoRD which use exact match and F1-measure. This is because even though the task is cloze, the answer should be generated (in case of CliCR) or extracted (ReCoRD). For extractive and generated tasks it is common to report exact match and F-1 measure. In the case of SearchQA, the authors investigate separately the length of the answers and report the accuracy for answers of certain fixed lengths as well as an exact match for answers of all lengths. CliCR, ReCoRD, MultiRC, Quasar, HotpotQA, QuAC and all extractive datasets use F-1 measure.

One can view MRC task from the perspective of Information Retrieval providing a ranked list of answers instead of one defined answer. In this case a Mean Reciprocal Rank (Craswell, 2009) (MRR) and Mean Average Precision (MAP) can be used, as well as the accuracy of the top hit (Hits@1) (single answer) over all possible answers (all entities). Assuming a system returns a list of answers, MRR is calculated as an averaged sum of reciprocal ranks, and a reciprocal rank of a question is the multiplicative inverse of the rank of the first correct answer. MAP is a mean of the average precision scores for each question. MRR and MAP are used only by Yang et al. (2015) in the WikiQA dataset, as well as P, R, F-1 measures. Miller et al. (2016) in the WikiMovies datasets used the

[36]Except MultiRC as there are multiple correct answers and all of them should be found

accuracy of the top hit (Hits@1).

All metrics mentioned above work well for well defined answers but might not reflect performance for those datasets where the original text has to be generated as there could be several alternative ways to answer the same question. Some datasets provide more than one gold answer. That is why a number of different metrics are used in addition: Bilingual Evaluation Understudy Score (BLEU) (Papineni et al., 2002), Recall Oriented Understudy for Gisting Evaluation (ROUGE-L) (Lin, 2004), and Metric for Evaluation of Translation with Explicit ORdering (METEOR) (Lavie and Agarwal, 2007). Those metrics take into account n-gram overlap and give credit to different answer possibilities. MSMarco, NarrativeQA, and TweetQA use BLEU, ROUGE-L, and METEOR along with MRR because they have generative answer and those metrics are more appropriate.

Choi et al. (2018) also introduce the human equivalence score (HEQ). It measures the percentage of examples where the system F1 matches or exceeds human F1, implying a system's output is as good as that of an average human. There are two variants: HEQ-Q is based on questions, i.e. the percentage of questions for which the system F1 matches or exceeds human F1; HEQ-D is based on dialogues, i.e. the percentage of dialogs for which the system F1 matches or exceeds human F1 for every question in the dialog.

2.6.2 Human Performance

Human performance figures have been reported for some datasets – see Table 2.2. This is useful in two ways. Firstly, it gives some indication of the difficulty of the questions in the dataset. For example, compare the low human performance score reported for the Quasar and CliCR datasets with the very high scores for DREAM, DROP and MCScript. Secondly, it provides a comparison point for automatic systems, which may serve to direct researchers to under-studied datasets where the gap between state-of-the-art machine performance and human performance is large, examples include CliCR (33.9 vs. 53.7), RecipeQA (29.07 vs 73.63), ShaRC (78.3 vs 93.9), HotpotQA (82.20 vs 96.37).[37]

[37]The sources of SOTA (all links last verified June 2020):
`paperswithcode.com/sota/question-answering-on-clicr`,
`hucvl.github.io/recipeqa`,

Although useful, the notion of human performance is problematic and has to be interpreted with caution. It is usually an average over individual humans, whose reading comprehension abilities will vary depending on age, ability to concentrate, interest in and knowledge of the subject area. Some datasets (CliCR, Quasar) take the latter into account by distinguishing between expert and non-expert human performance, while RACE distinguishes between crowdworker and author annotations. The authors of MedQA (Zhang et al., 2018b), which is based on medical examinations, use a passing mark (of 60%) as a proxy for human performance. It is important to know this when looking at its "solved" status since state-of-the-art accuracy on this dataset is still only 75.3% (Zhang et al., 2018b).

Finally, Dunietz et al. (2020) call into question the importance of comparing human and machine performance on the MRC task and argue that the questions that MRC systems need to be able to answer are not necessarily the questions that people find difficult to answer.

2.7 What makes a dataset difficult?

In previous sections of this chapter I consider various properties of datasets. In this section I investigate whether there is any correlation between those properties and both human and state-of-the-art performance for those datasets. This is an attempt to answer the question *"What makes a dataset difficult?"*

For this analysis I selected those datasets where the human performance is reported and measured in accuracy. There are 22 datasets: bAbI, BoolQ, CBT, DREAM, MCScript, MCScript2.0, MCTest 160/500, MovieQA, PubMedQA, QAngaroo WikiHop, Quasar-S, Quasar-T, RACE, Recipe QA, ReCoRD, ShARC, TriviaQA, and WhoDidWhat.

I selected the following features of the datasets for analysis:

- **Popularity:**[38] how popular is the dataset measures by the number of citations from

`sharc-data.github.io/leaderboard.html`,
`hotpotqa.github.io`

[38]Data collected and last verified April 2021.

Google Scholar[39] and number of teams on a dataset leaderboard.

- **General Statistics** which include number of questions/passages, average length of passage/question/answer, full vocabulary size, number of all named entities, and number of unique named entities;

- **Domain** is represented as a boolean feature and indicates whether a dataset belongs to a particular domain, as discussed in Section 2.3;

- **Creation method** which is represented as five boolean features: automatically generated, human-generated, user-generated, crowdworkers, and usage of knowledge graph. Each of them distinguishes between whether a dataset was created with the the indicated method method or not. For example, if a dataset was create using crowdworkers and a knowledge graph, those two features would be represented as "1", and the rest as "0";

- **Additional features** which are also represented as binary features such as: availability of boolean questions, non-factoid questions, questions in a query form, questions required multi-hop reasoning, multiple documents in the passage, questions without an answer, and any additional data, which were discussed in Section 2.3.3.

2.7.1 Human Performance and SOTA Regression

First, I applied a regression model to the individual features and selection of features described above to predict the human performance and SOTA. The goal of this experiment is not to build a model which can predict human and SOTA performance but to identify which of those datasets' properties have a significant effect on the performance. To do so I used Linear Regression (Ordinary Least Squares) from the `statsmodels` library.[40]

[39] https://scholar.google.com/ – last verified May 2021.

[40] `https://www.statsmodels.org/stable/index.html` – last verified May 2021. The method `statsmodels.api.summary` returns the summary of the model including coefficients and p-value. A sign of the coefficient indicates whether the correlation is positive or negative, while the p-value indicates significance. More information can be found here: `https://medium.com/swlh/interpreting-linear-regression-through-statsmodels-summary-4796d359035a` – last verified November 2021

	Individual Features	Additional Paired Features
HP	- average question length; - average passage length; - automatic generation; - non-factoid questions; - multi-hop reasoning; - Story domain;	- average question length and News domain; - average passage length and Medical domain; - number of unique NE and multiple documents; - crowdsourcers and Medical domain; - Medical and Other domains.
SOTA	- number of citations; - number of teams; - average passage length; - automatic generation; - multi-hop questions.	- number of teams and other domain; - number of questions and average question length; - number of questions and multi-hop passage; - average passage length and Medical domain; - multi-hop passage and Medical domain; - Medical and Other domain.

Table 2.5: Statistically significant properties for Human Performance and SOTA. Features with negative correlation are marked red.

First I consider the human performance (HP) as an independent variable and look into all other properties individually. I also repeat the experiment with all possible combination of paired properties. Then I repeat the same experiment with SOTA as an independent variable. The results which are significant at $p < 0.05$ are shown in Table 2.5.

2.7.2 Feature Correlation

Next, I look into correlation between features applying Pearson's correlation. To simplify the visual understanding of results in this section I marked the following: positive strong correlation ($r > 0.5$) with a green cell, and positive medium correlation ($0.3 < r < 0.5$) with a yellow cell.

The results for HP, SOTA, number of citations, and number of teams on the dataset leader board is presented in Table 2.6.

Unsurprisingly, SOTA strongly correlates with HP and the number of teams on a dataset leaderboard, as if questions are difficult for people they might also be difficult for systems as well, and the greater the number of participants, the greater the level of competition. There is also a medium correlation between SOTA and the number of citations, which is reasonable as citations indicate that a dataset is being used.

The results of correlation for all other features are presented in Table A.7 in the Ap-

	HP	SOTA	# citations	# teams
HP	-	0.73	0.26	0.25
SOTA		-	0.43	0.51
# citations			-	0.15
# teams				-

Table 2.6: Correlation for performance (**HP** and **SOTA**) and popularity (number of citations and number of teams on the leader board) of dataset.

pendix. Here are some correlations that I observe:

- There is a strong positive correlation between the number of passages (also passage length) with the size of the vocabulary, and the number of all and unique NEs. This make sense as the longer the text, the more words are in it, and the higher the probability of getting different words;

- HP has a strong negative correlation with the average passage length which is logical, as longer text might be harder to comprehend;

- HP has a medium positive correlation with crowdsourcers, and boolean questions. It suggests that those questions are easier for humans to answer;

- HP has a medium positive correlation with non-factoid questions. This might look a bit surprising but also reasonable as people are generally good with answering non-factoid questions, e.g. see HP for HotpotQA (Yang et al., 2018b), DREAM (Sun et al., 2019), and MCScript2.0 (Ostermann et al., 2019);

- Human generated samples (either questions, or passages, or both) have a medium positive correlation with the average answer length, vocabulary, and the number of all NE. This makes sense, as when not framing questions for a particular task, people might express themselves with longer statements which also leads to richer vocabulary;

- Both HP and SOTA negatively correlate with automatically generated data. This suggests that automatically generated data is harder to understand;

- Both HP and SOTA have a strong positive correlation with multi-hop reasoning. This suggests people are generally able to reason over multiple sentences. It also imply that the multi-hop reasoning is not so challenging;

- Both HP and SOTA have a medium negative correlation with the Medical and Other domains. This suggests that those domains are more difficult compared to other MRC datasets;

- Datasets created by crowdsourcers have a strong negative correlation with the question length and a medium negative correlation with the number of passages and the passage length. This point make sense as within a defined task crowdsourcers create short and precise text. It is also possible, the crowdsourcers are given tasks with shorter passages.

2.7.3 Questions and Answers for Humans and MRC

Another observation I find important to mention is the difference between those datasets which were created for machine reading comprehension specifically, and those which were created originally for people and then were reused in machine reading comprehension (such as exam questions, quizzes, and questions from user fora).

Taking the same subset of datasets from the previous section where the full number of named entities were calculated I classified them as described above. The majority of those datasets were created specifically for machine reading comprehension while, only 5 out of 20 datasets were reusing questions created originally for people: BoolQ, Quasar-S, Quasar-T, RACE, and TriviaQA.

The difference between human performance and SOTA is illustrated on Figure 2.8 (a-b). For datasets based on questions for humans, both average HP and SOTA are around 70% accuracy, while, for datasets created for MRC, HP is over 75% and SOTA is 90% accuracy.

Figure 2.8 (c) reflects the dramatic difference in average passage length, which has a strong negative correlation with both HP and SOTA, as was shown in the previous section.

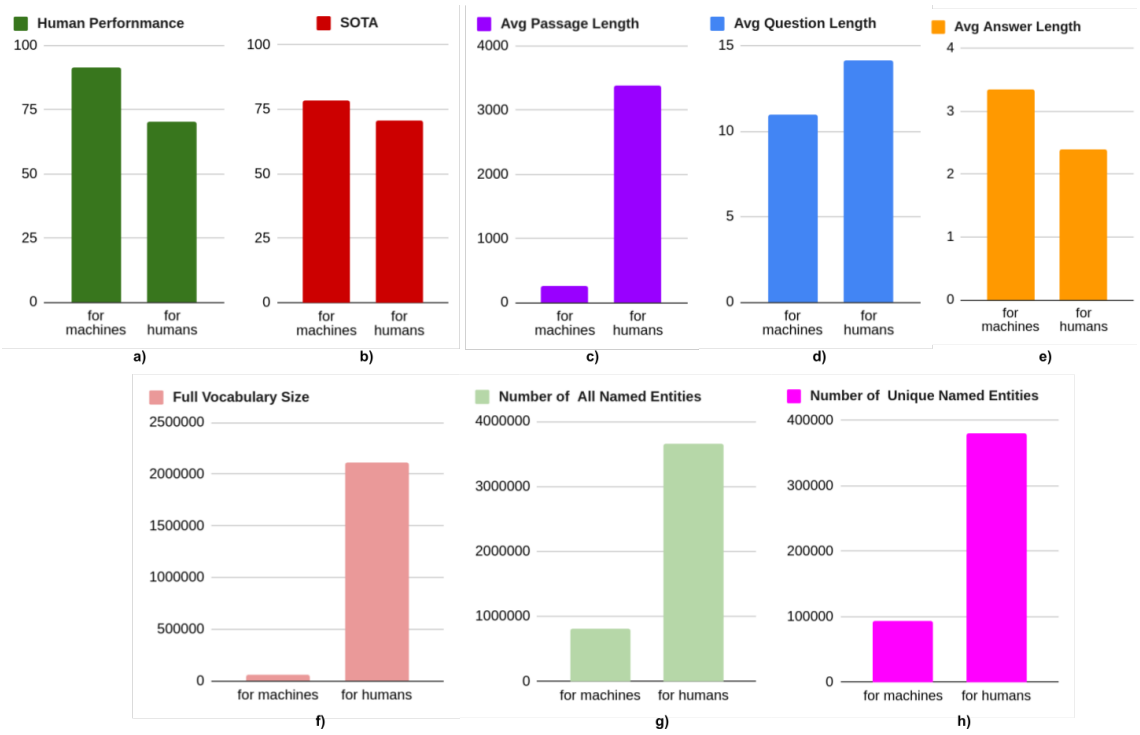


Figure 2.8: Difference between datasets for MRC and those which reused questions for people. (a) average human performance, (b) average SOTA, (c) average passage length, (d) average question length, (e) average answer length, (f) average vocabulary size, (g) average number of all named entities, and (h) average number of unique named entities.

Additionally, Figure 2.8 (d, e) shows the difference in average question and answer length. The questions for people are slightly longer while the answers tend to be a bit shorter. Figure 2.8 (f-g) demonstrates the difference in the average number of all named entities, average number of unique named entities, and average vocabulary size. The number of all named entities and unique named entities is four times bigger for those datasets which were originally aimed at people. The difference in average vocabulary size is very large.

This is a rough comparison and could be investigated further with a larger amount of datasets. However, it suggests that those datasets which are based on questions designed for people are more difficult to solve than those which were created specifically for the MRC task.

2.8 Summary

In this chapter, I have presented an up-to-date, one-stop-shop survey of 54 English reading comprehension datasets. I compare the datasets by question and answer type, size, data source, creation method, vocabulary, question type, "solvedness" and human performance level. Seeing the history of dataset creation, I can observe the tendency of moving from smaller datasets towards large collections of questions, and from synthetically generated data through crowdsourcing towards spontaneously created. I also observe a lack of *why*, *when* and *where* questions.

Gathering and processing the data for this survey was a painstaking task, from which I emerge with some very practical recommendations for future MRC dataset creators. In order to 1) compare to existing datasets, 2) highlight possible limitations for applicable methods, and 3) indicate the computational resources required to process the data, some basic statistics such as average passage/question/answer length, vocabulary size and frequency of question words should be reported; the data itself should be stored in a consistent, easy-to-process fashion, ideally with an API provided; any data overlap with existing datasets should be reported; human performance on the dataset should be measured and what it means clearly explained; and finally, if the dataset is for the English language and its design does not differ radically from those surveyed here, e.g. the recent Template of Understanding approach of Dunietz et al. (2020), it is important to explain why the field needs this new dataset.

The study contributes to the field as follows:

1. it describes and teases apart the ways in which MRC datasets can vary according to their question and answer types;
2. it provides analysis in a structured and visual form (tables and figures) to facilitate easy comparison between datasets;
3. by providing a systematic comparison, and by reporting the "solvedness" status of a dataset, it brings the attention of the community to less popular and relatively understudied datasets;

4. per-dataset statistics such as number of instances, average question/passage/answer length, vocabulary size and text domain can be used to estimate the computational requirements for training an MRC system.
5. analysis shows that for all domains except *Product* and *Exams* the absolute majority of questions and answers contain *Person* type named entities, while for the *Product* domain, as expected the most common named entities are *Misc*, which includes some actual product names; also for the *Exam* domain the types of named entities are distributed relatively equally across answers;
6. additionally I perform the analysis of the datasets complexity by analysing significance of regression coefficients for individual and paired properties and providing correlation between them. I showed that those datasets which were created specifically for MRC are easier than those which reuse questions created for people.

The result of this work has been published at The 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP),[41] (Dzendzik et al., 2021).

I open source my code for processing datasets and additional analysis for everyone who would like to use it.[42]

[41] `https://aclanthology.org/2021.emnlp-main.693/` – last verified November 2021
[42] `https://github.com/DariaD/RCZoo` – last verified June 2021

Chapter 3

Reading Comprehension Methods

> It's still magic even if you know how it's done.
>
> *Terry Pratchett*
>
> *A Hat Full of Sky*

In this chapter, I provide a detailed description of the methods which are necessary to understand this work and the current state-of-the-art architectures for reading comprehension. The majority of selected approaches in my work were driven by the state-of-the-art results at the time and usability concerns.

There are some other works: Zhang et al. (2019); Liu et al. (2019a); Qiu et al. (2019a); Baradaran et al. (2020); Wang (2020); Jurafsky and Martin (2020) which also describe the most common models for RC.

Firstly, I start with a number of word embedding representations (Section 3.1). Secondly, I discuss text similarity measures (Section 3.2). Thirdly, I provide a detailed description of Logistic Regression (Section 3.3). Both, string similarity and logistic regression I use in Chapter 4, and string similarities are used in Chapter 6. Fourthly, I give an introduction to some neural network architectures which are widely used in reading comprehension and question answering (Section 3.4), such as Recurrent Neural Networks and Long Short-Term Memory (Hochreiter and Schmidhuber, 1997), Sequence to Sequence models (Sutskever et al., 2014), Attention (Bahdanau et al., 2015). It is important to un-

derstand those architectures as they were widely used in related work and form a basis for more advanced methods. After that, I provide a description of the Transformer architecture introduced by Vaswani et al. (2017) and its modifications such as OpenAI Transformer (Radford et al., 2018, 2019), BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019b) (Section 3.4.4). I use and analyse the transformer-based architectures in Chapters 5-7 I then talk about Graph Neural Networks (Scarselli et al., 2009) (Section 3.4.5) which I use in Chapter 6. Finally, I conclude this chapter with a summary in Section 3.5.

3.1 Word Embedding Representation

Humans perceive words as a sequence of symbols and the concepts behind them. Those concepts are based on life experiences.

There are two sentences in (3.1). Humans understand that *desk* and *table* are the same or very similar things, even though the symbolic representations are very different. The key idea of the *word embedding* is words which are semantically similar likely to appear in a similar context and should have a similar representation.

(3.1) Pandas sit on the desk. Pandas sit on the table.

To help systems "understand" actual natural language and to encode similarity, words should be transformed into numerical representations. This process is called *word embedding* and it is an important component of text understanding due to the fact that it is capable of reconstructing some semantic connections, as well as syntactic, or grammar-based relationships (Collobert and Weston, 2008; Collobert et al., 2011). Obtaining a good word representation becomes a task in itself with the goal to capture as much of the semantic, morphological, hierarchical, etc information from the word as possible.

Initially, word embeddings were based on word frequency (Miller and Charles, 1991). Then it was common practice to learn word embeddings alongside models on each particular dataset, e.g. a distributed representation for words proposed by Bengio et al. (2003). After that, the word embedding models were trained on a huge number of texts such as

books/news corpora and Wikipedia. Those models became available for use in different NLP tasks (Collobert and Weston, 2008; Collobert et al., 2011; Mikolov et al., 2013b; Pennington et al., 2014). I consider a number of semantic word representations categorised by type: *frequency-based*, *static*, and *contextual*. While I keep the description of word embedding limited to what is required to understand this work, Baroni et al. (2014) provide a detailed comparison of the predictive models with count-vector-based distributional semantic approaches and show the superiority of the first method.[1].

3.1.1 Frequency-based Word Embedding

3.1.1.1 Count Vector

The first frequency-based word embedding is **count vector**. It is a numerical representation of words through the one-hot encoding method. The idea is to create a vector that has as many dimensions as a number of unique words in the dictionary, where *dictionary* is a set of all words used in the number of considered documents. Each unique word is represented as 1 in a unique dimension (index) and the rest is filled with 0s, e.g. Example (3.2) for two documents which has a dictionary of 6 words. This method is straightforward but it does not capture any semantic information.

(3.2) **Document 1** Pandas are lazy. **Document 2** Pandas have six toes.

Word Embedding:

are	[1 0 0 0 0 0]	have	[0 1 0 0 0 0]
lazy	[0 0 1 0 0 0]	pandas	[0 0 0 1 0 0]
six	[0 0 0 0 1 0]	toes	[0 0 0 0 0 1]

The document can be represented as a count of words e.g. a vector with the dimension of dictionary size where the value of the vector in every word position shows how many

[1]There is a whole literature in terms of distributional semantic models between TF-IDF and static word embeddings, such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003), bound encoding of the aggregate language environment (BEAGLE) (Jones and Mewhort, 2007), and Hyperspace Analogue to Language (HAL) (Lund et al., 1995; Lund and Burgess, 1996)

times the word appears in the document (rows in Example (3.3)). And words can be presented by frequency in the document (columns in Example (3.3)).

(3.3) **Document 1** Pandas are lazy.

Document 2 Pandas have six toes.

Document 3 Pandas are lazy and pandas are cute.

Embedding:

	and	are	cute	have	lazy	pandas	six	toes
Document 1	0	1	0	0	1	1	0	0
Document 2	0	0	0	1	0	1	1	1
Document 3	1	2	1	0	1	2	0	0

3.1.1.2 TF-IDF

The idea of using **Term Frequency** (**TF**) comes from the task of ranking documents according to the relevance to a query. The number of times each term (word) from the query occurs in each document can be counted and is called its term frequency.

For long documents the weight of the term can be adjusted accordingly (Luhn, 1957):

$$TF(t,d) = \frac{n_t}{\sum_{k=1}^{K} n_k} \tag{3.4}$$

where n_t is a number of times the term t appears in the document d and $\sum_{k=1}^{K} n_k$ is a sum of all term frequencies in the document (which is equal to the number of all non-unique words in the document).

The term frequency tends to give words such as *and* or *the* too much weight, without giving enough weight to the more meaningful terms in the query. To address that Spärck Jones (1972) introduced **Inverse Document Frequency** (IDF) based on the idea of the importance of the words which are common for the relevant document but not so frequent in other documents. See Equation (3.5).

$$IDF(t,D) = log\frac{|D|}{|\{d_i \in D | t \in d_i\}|} \tag{3.5}$$

where D is a collection of documents, $|D|$ is the number of documents in the collection and $|\{d_i \in D | t \in d_i\}|$ is the number of documents where $n_t > 0$.

Those two ideas combine powerful characteristics of the words in the context of the document: term frequency and inverse document frequency (**TF-IDF**) which is the product of two statistics (see Equation 3.6).

$$TF\text{-}IDF(t, d, D) = TF(t, d) * IDF(t, D) \tag{3.6}$$

For example, considering the same documents from (3.3) above the words *pandas* and *toes* have the term frequency, inverse document frequency, and a combination of those as shown in (3.7).

(3.7)

	pandas			toes		
	TF	IDF	TF-IDF	TF	IDF	TF-IDF
Document 1	$\frac{1}{3} = 0.34$	$log\frac{3}{3} = 0$	0	$\frac{0}{3} = 0$	$log\frac{3}{1} = 0.48$	0
Document 2	$\frac{1}{4} = 0.25$	0	0	$\frac{1}{4} = 0.25$	0.48	**0.12**
Document 3	$\frac{2}{7} = 0.29$	0	0	$\frac{0}{7} = 0$	0.48	0

The word *panda* is very common for this collection of documents and even though TF is positive (0.34, 0.25, and 0.29) TF-IDF is 0, while for the word *toes*, as it is mentioned only in the second document, the TF-IDF score is 0.12 for this document and the score is 0 for the other documents.

3.1.2 Static Word Embedding

The next step regarding word embedding has been done with the use of unsupervised learning where the model assigns a vector value to the words by processing a huge volume of texts. The *static word embedding* is a function which maps each word to a numerical vector. Vector dimension is much smaller than the number of words in the vocabulary. The biggest differences with the count vectors is the fact that this representation is able to capture semantic difference between the words. The representation is called *static* as regardless of the different meanings of the same word, the same vector will be applied.

Word2Vec, proposed by Mikolov et al. (2013a,b), is a number of word embedding models. They aimed to find a way to learn a good quality vector world representation from huge data sets (vocabulary size $>$ 1000000 words) which would capture the semantic similarity of words. The models are trained to reconstruct the linguistic contexts of words.

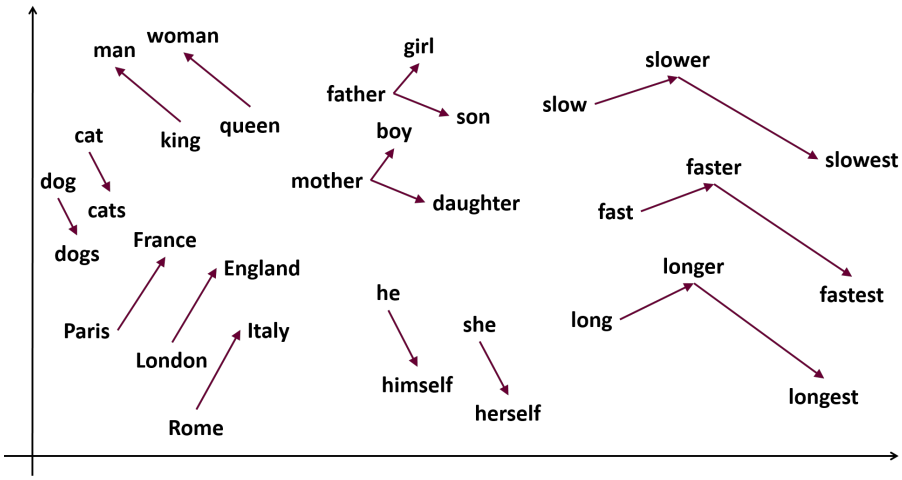


Figure 3.1: Illustration of word2vec vectors in space. Picture credit: `https://samyzaf.com/ML/nlp/nlp.html` – last verified December 2021

The main idea is that vectors are positioned in the vector space so that words that share common contexts in the corpus are located in close proximity to one another in the space (see Figure 3.1). For example, the distance between words *Paris* and *France* would be the same (or close enough) to the distance between words *Rome* and *Italy*. Or another classical example of Word2Vec is: *king* - *man* + *woman* $\approx$ *queen*

There are two variations of the *word2vec* algorithm: **Continuous Bag Of Words** (CBOW) which learns to predict the word based on the context around it, and **Skip-Gram** which is the other way around: learn to predict the context given the focus on a central word. The Figure 3.2 from Mikolov et al. (2013a) shows the difference between CBOW and Skip-Gram approaches.

Word2Vec is implemented as a two-layer neural network which takes as its input each context word (CBOW) or each target word (Skip-gram) as a one-hot vector from a large corpus of documents, maps each one-hot vector to a dense d-dimensional vector, averages these vectors, and predict the target word (CBOW) or predict the context word (Skip-

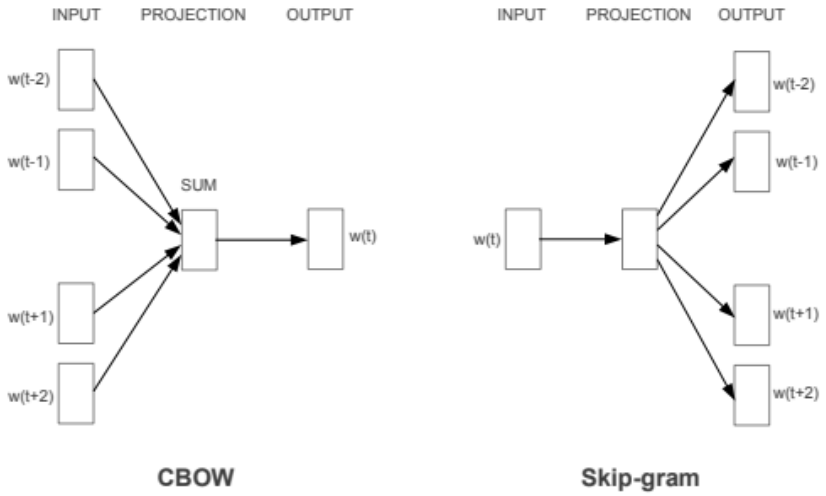


Figure 3.2: Model architectures from Mikolov et al. (2013a). The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts the surrounding words given the current word.

gram) with a $softmax : \mathbb{R}^{|K|} \rightarrow \mathbb{R}^{|K|}$ function which is defined in Equation 3.8:

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \tag{3.8}$$

where K is the dimension of input vector z.

There is a number of other static word embedding algorithms which I did not use in my experiments, inclusing **GloVe** (Pennington et al., 2014) and **FastText** (Bojanowski et al., 2017).

3.1.3 Contextual Word Embedding

The limitations of the static word embeddings are lexical ambiguity and context understanding, as every word always has the same static vector representation in every context. For example, in (3.9-3.11) the words *panda* and *pandas* stand for three completely different things: the first is an animal, the second is a software library, and the third one is waste and recycling service. With the static word embedding all three instances would have the same vector representation regardless of the context.

(3.9) Giant pandas are good at climbing trees and can also swim.

(3.10) Pandas is an open source data analysis tool.

(3.11) Panda - Collecting your bins every week is our passion.[2]

Peters et al. (2018) introduced deep contextualized word representation called **ELMo** (Embeddings from Language Models). It provides different representations depending on the sentence context. To do so, ELMo uses a bi-directional Long-Short Term Memory Network (biLSTM) (LSTM will be discussed in Section 3.4.1.1) trained on next word prediction language modeling tasks.

Another approach for contextual word representation has been introduced with transformer architecture. I will discuss this in detail in Section 3.4.4.

3.1.4 Sentence Representation

The idea of word embedding quickly moved from a word level to a sentence and paragraph level, e.g. **doc2vec** (Le and Mikolov, 2014). I provide some details of two approaches as I use them for a multiple choice QA system in Chapter 4 and answering boolean questions in Chapter 6.

Skip-thoughts proposed by Kiros et al. (2015) - is a model which is trained on the text from books and represents semantic and syntactic information. According to Kiros et al. (2015), the model encodes a passage and tries to reconstruct the surrounding sentences. So, sentences with similar semantic and syntactic properties are mapped to similar vector representations. In other words, skip-thought uses a similar approach to word2vec but on a sentence level rather than a word level. To train the model they used encoder-decoder architecture proposed by Cho et al. (2014): the encoder is a Recurrent Neural Network (RNN) with Gated Recurrent Units (2 gates) and decoder is another RNN. I will talk about this architecture in more detail in Section 3.4.

[2]Panda is a waste and recycling service in Dublin. The sentence is taken from `https://www.panda.ie/household/` – last verified January 2021

Conversational based sentence-level representation is proposed by Yang et al. (2018a) and used by me in Chapter 6. The model is trained to predict responses on Reddit conversations (Al-Rfou et al., 2016) and supplemented with natural language inference task (SNLI) (Bowman et al., 2015) to learn representations for sentence-level semantic similarity. They note a semantic similarity between responses to semantically similar questions. The task is formulated as a response prediction: the correct response should be selected from a number of randomly selected responses. The model encodes the input and all responses into fixed-length vectors and scores all input-response pairs as the dot product of the two vectors. Then the *softmax* function is applied to receive probabilities for all responses and the model maximizes the log-likelihood of the correct responses.

Yang et al. (2018a) experiment with a number of encoder architectures and empirically established that the transformer-based encoder works the best.

3.2 String Similarity

The idea of using text similarity in the Reading Comprehension Question Answering task is intuitive. When looking for the answer, I would like to see the part of a passage that is the most "similar" to a question and then select/extract/generate the answer.

In this section I will talk about the different types of string similarities in general while putting the reading comprehension task slightly aside. More practical discussions about the specific usage of string similarities in question answering will be presented in future chapters when I will introduce my approach.

Every similarity is a function which takes two strings as arguments: Str_1 and Str_2. The work of Gomaa and Fahmy (2013) considers more than 25 text similarities, divided into five groups: character-based similarity, term-based similarity, corpus-based similarity, knowledge-based similarity, and hybrid similarity measures. I consider the following simple similarities: `Cosine`, `WindowSlide`, `BagOfWords` and `N-grams`.[3]

Cosine similarity is a widely used way to measure the string similarity. First it is

[3]Those metrics are selected as it seems to be natural to use them, and also they were used in the related work at the time, particularly by Rajpurkar et al. (2016) and Tapaswi et al. (2016).

necessary to convert the string into vectors and calculate the cosine similarity between two vectors as Equation (3.12) shows:

$$w2v\_cos(Str_1, Str_2) = \frac{v_{Str_1} \cdot v_{Str_2}}{|v_{Str_1}||v_{Str_2}|} \tag{3.12}$$

where v_{Str_1} and v_{Str_2} are some vector representations of two strings, as discussed in Section 3.1.

Bag of words ratio similarity – a bag of words measure shows the ratio of the words from the string Str_1 which exists in the other string Str_2 to the length of the first string as shown in Equation (3.13):

$$bow(Str_1, Str_2) = \frac{|W_{Str_1} \cap W_{Str_2}|}{|W_{Str_1}|} \tag{3.13}$$

where W_{Str_1} is the bag of words from the string Str_1 and W_{Str_2} the bag of words from the string Str_2.

Window slide ratio similarity – returns the highest ratio of sequence match between the first string and all substrings of the second string. The window of the substrings has a size equal to the length of the first string. See Equation (3.14):

$$wSlide(Str_1, Str_2) = max_i(\frac{2 * M_i}{|Str_1| + |s_i|}) \tag{3.14}$$

where $|Str_1| + |s_i|$ is the total number of character elements in both sequences, the Str_1 and s_i, where s_i is i–substring of Str_2, $\forall i, |s_i| = |Str_1|$ and M_i is the number of matches between all substrings of Str_1 and s_i.

The idea of window slide is also described in Richardson et al. (2013); Hill et al. (2016); Rajpurkar et al. (2016)

Character N-gram – is similar to `Window Slide` but works on a character level. The size of the window is limited by the parameter n ($n = 2, 3, 4, 5$ characters is considered in this work).

3.3 Logistic Regression

A number of tasks can be formulated as a task of classification, including reading comprehension question answering. Boolean question answering can be considered to be binary classification, while multiple choice question answering is defined as multi class classification.

Before moving towards more complex architectures, I would like to describe the algorithm of simple linear classifiers such as logistic regression, which I use in Chapter 4. It was developed as a model of population growth and named "logistic" by Pierre François Verhulst in the 1830s and 1840s, and is based on the logistic function. Even though it was called *regression*, it is commonly used for classification.

3.3.1 Binary Classification

The classic application of logistic regression is binary classification, e.g. to divide emails into spam and not spam, or to identify the sentiment of reviews as positive or negative.

Let me denote the input to the classifier as x (a vector of input features) and the output as y (a class label) and consider binary classification (means $y \in \{0, 1\}$). Then y is calculated as a probability P with the condition x (see (3.15)).

$$\hat{y} = \begin{cases} 0 & \text{if } P(y = 1|x) < 0.5 \\ 1 & \text{if } P(y = 1|x) >= 0.5 \end{cases} \tag{3.15}$$

Logistic regression assumes the instances can be separated in a vector space by a linear function, so the linear hypothesis $z = Wx + b$ take place, where W is a matrix of weights, and b is bias vector, both will be learned during the training process. To fit the value of the linear hypothesis z into the $(1, 0)$ interval, the probability of $y = 1$ with input x $P(y = 1|x) = \psi(z)$ is calculated as a standard logistic function (see Fig. 3.3) (*sigmoid*) as defined in Equation (3.16).

$$\hat{y} = \psi(z) = \frac{1}{1 + e^{-z}}, \ \forall z \, \psi(z) \in (0, 1) \tag{3.16}$$

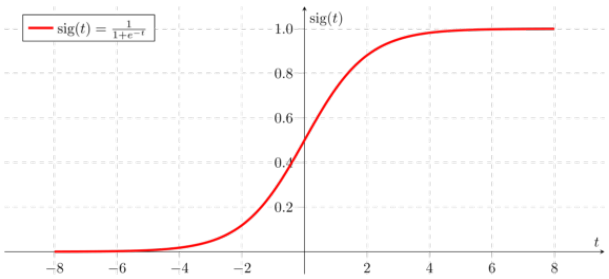


Figure 3.3: Sigmoid function. Picture credit: Saishruthi Swaminathan `https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc` – last verified January 2021

As I mentioned above, both W and b are parameters to be learned. In order to learn those parameters a cost function (loss function) has to be defined to measure the difference between the prediction $\hat{y}$ and golden label y.

The cost function is defined by the Equation (3.17) as $P(y = 1|x) = 1 - P(y = 0|x)$ and it is a high value if the prediction is incorrect and 0 if y and $\hat{y}$ are identical.

$$cost(y, \hat{y}) = \begin{cases} -log(\psi(z)) & \text{if } y = 1 \\ -log(1 - \psi(z)) & \text{if } y = 0 \end{cases} \tag{3.17}$$

Two conditions are combined together with the $cost$ function as shown on Equation (3.18).

$$cost(\hat{y}, y) = -y\, log(\psi(z)) - (1 - y)\, log(1 - \psi(z)) \tag{3.18}$$

Decreasing the cost will increase the maximum likelihood of predicting the correct labels.

3.3.2 Non-Binary Classification

The binary logistic regression can be used for non-binary classification in two ways: **one vs. rest** and **pairwise classification**. It is a set of classes K, the first approach involves training $|K|$ classifiers where $|K|$ is a number of classes. So each classifier is trained to assign a label of a particular class. The pairwise classification involves training the clas-

sifier for every pair of classes and the final decision is made by the majority of classifiers which detected the class. $|K-1|$ classifiers would provide the label for a new item with the confidence and the rest would assign the label randomly. The final label is obtained as a majority voting.

For multinomial (multi-label) non-binary classification with $|K|$ classes, instead of utilizing the sigmoid function ψ the *softmax* function $softmax : \mathbb{R}^{|K|} \rightarrow \mathbb{R}^{|K|}$ should be used, as now the probability of more than two classes has to be predicted. Now the linear hypothesis z is calculated for each class k: $z_k = W_k x + b_k$, $\forall k \in K$. For each class k the probability is calculated according to the Equation (3.19) where $k = 1, ..., K$ and $z = (z_1, ...z_K) \in \mathbb{R}^{|K|}$

$$P(y = k|x) = softmax(z)_k = \frac{e^{z_k}}{\sum_{k=1}^{K} e^{z_k}} \tag{3.19}$$

The final answer is calculated as $argmax$ which returns the vector $y \in \mathbb{R}^{|K|}$ with all zeros except for a single 1 for the class with the max probability. The index of the 1 is the answer class.

3.4 Common Neural Network Architectures

The first artificial neuron called *"perceptron"* was described by Rosenblatt (1958). This was followed by 40 years of fundamental study of artificial learning processes and combining neurons into more complex architectures, e.g. works of Widrow and Hoff (1962); Anderson (1972); Kohonen et al. (1977); Hopfield (1982); Reilly et al. (1982); Hinton et al. (2006). Since then neural networks have been used in many fields including text understanding.

A group of perceptrons (units) or *neurons* connected to the input form a layer of a neural network. There could be many layers on the top of each other. In other words, perceptron is a one-unit one-layer network. The neural network where each layer is passing information from the previous layer forward without loops and connections between units of the same level is called a *feedforward network*. Normally, each unit of the current

layer is receiving information from every unit of the previous layer and passing to each unit of the following layer. A perceptron is illustrated on Figure 3.4 (a), and Figure 3.4 (b) illustrates a simple feedforward network, which contains one input layer, two hidden layers and one output layer, but the input layer is not counted, so this is a 3-layer network.

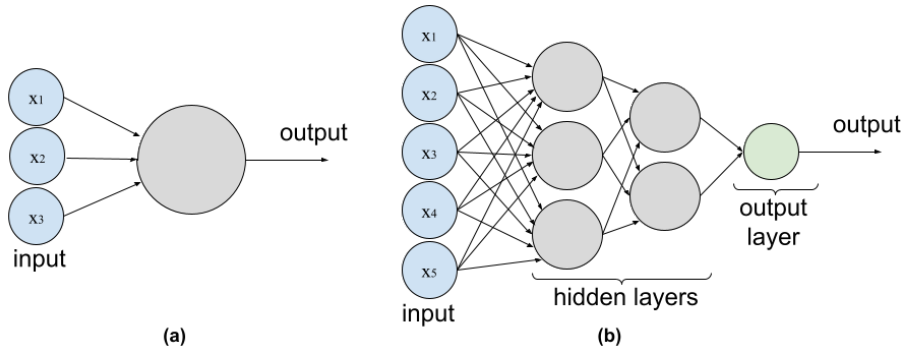


Figure 3.4: (a) The perceptron layer and (b) Multilayer feedforward neural network.

3.4.1 Recurrent Neural Networks

Recurrent Neural Network (RNN) is a widely used model for NLP tasks. The main advantage of an RNN is its ability to process a sequence with a memory of elements which have been processed so far. On the first step, the first element of the sequence is fed to the RNN. On the second step, the second element and the result of the processing of the first element are fed into the RNN. So for every next step, the net will "remember" what has been processed before because the result of every current step is a part of the input. To do so, RNN uses a memory (their internal state). Figure 3.5 shows an unrolled recurrent neural network.

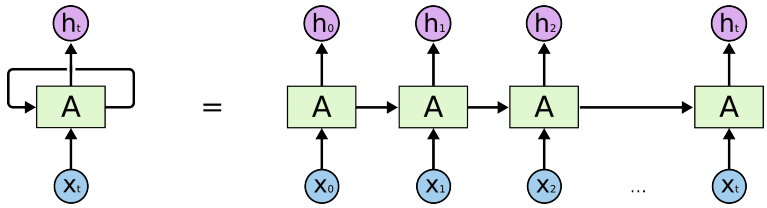


Figure 3.5: An unrolled recurrent neural network. Picture credit: Christopher Olah `https://colah.github.io/posts/2015-08-Understanding-LSTMs/` – last verified December 2021

3.4.1.1 Long Short-Term Memory Network

RNNs tend to suffer from a vanishing gradient problem or, the opposite, an exploding gradient problem. That happens during training when the weights should be updated but the gradient is extremely small (or huge). That can cause a stop in the model training. One possible solution was introduced by Hochreiter and Schmidhuber (1997) and is called a Long Short-Term Memory network (LSTM).

LSTM is a powerful modification of RNN. Every neuron of the net has three gates: input gate, output gate and forget gate. The gates help to control the information that is going from the previous step to the current step and the current step to the next step. This approach helps to learn the net in a more efficient way.

3.4.1.2 Encoder-Decoder Architecture and Sequence to Sequence Model

The encoder-decoder architecture based on RNN was proposed by Cho et al. (2014) for Statistical Machine Translation. As can be concluded from the name, it contains two parts: the **encoder** is a network which transforms the input sentence into a fixed-length vector, and the **decoder** is a network which transforms the vector back to variable length sequence (see Figure 3.6). Cho et al. (2014) proposed to use the LSTM as encoder and decoder. Any network can be used.

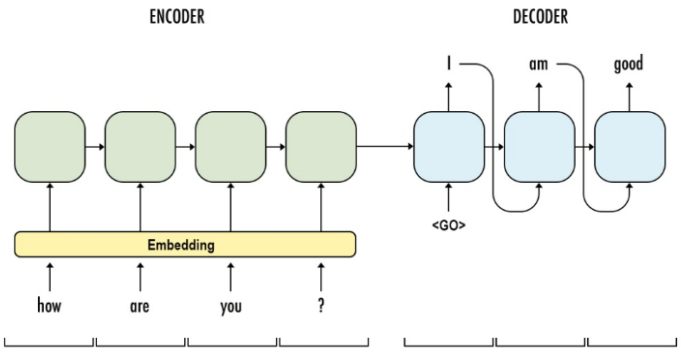


Figure 3.6: Encoder-decoder architecture. Picture credit: Chaoran
`https://6chaoran.wordpress.com/2019/01/15/build-a-machine-translator-using-keras-part-1-seq2seq-with-lstm/`
– last verified September 2020

In general, text is a sequence of words or characters. In other words, the answer gen-

eration task can be described as a task of producing one sequence (answer) from another sequence (question and passage).

Sequence-to-sequence (seq2seq) is an end-to-end model based on LSTM proposed by Sutskever et al. (2014). It is a generalised idea of end-to-end learning with input and target sequences. Originally, this model was applied to the machine translation tasks. This approach has become widespread in other fields of natural language generation including a reading comprehension task. The main idea is: one LSTM model is used for encoding a sequence of any length to a fixed dimensional vector and another LSTM is used for decoding the vector to a target sequence. The authors show that deep LSTM with four layers works better than shallow LSTM (Sutskever et al., 2014, p.3). Again, generally speaking, those two networks do not have to be LSTMs, but the LSTMs were used in the original paper and this architecture is still widespread.

3.4.2 Attention Mechanism

The idea of attention comes from paying attention to word alignment in the Neural Machine Translation (NMT) task (Bahdanau et al., 2015). With long sequences, the model tends "to forget" the beginning when it get to the end of a sequence. Instead of forward (or backward) feeding of input, the attention mechanism allows feeding a different part of the source sequence at each step. It helps the model learn what "to attend to" next based on the previously processed part of the input sequence.

Formally, $x = (x_1, x_2, ..., x_n)$ is a input sequence and $y = (y_1, y_2, ..., y_m)$ is a output sequence.

Encoder: The encoder state h_i is a concatenation of forward hidden state $\vec{h_i}$ and backward hidden state $\overleftarrow{h_i}$ as shown on Equation (3.20).

$$h_i = [\vec{h_i}^\top; \overleftarrow{h_i}^\top]^\top, i = 1, ..., n \tag{3.20}$$

Decoder: Then the hidden state s_t of decoder for y_t (the output word at position $t, t = 1, ..., m$) is described by Equation (3.21):

$$s_t = f(s_{t-1}, y_{t-1}, c_t) \tag{3.21}$$

where s_{t-1} is a hidden state for the previous word, and c_t is the context vector calculated as a sum of weighted hidden states of the input sequence (see Equation (3.22)).

$$c_t = \sum_{i=1}^{n} \alpha_{ti} h_i \tag{3.22}$$

The alignment score α_{ti} shows the alignment between two words x_i and y_t. It is computed according to Equation (3.23) for every hidden state h_i:

$$\alpha_{ti} = \frac{e^{a(s_{t-1,h_i})}}{\sum_{k=1}^{n} e^{a(s_{t-1,h_k})}} \tag{3.23}$$

where a is an *alignment model* and it shows a score of how close the inputs around position i and the output around position t match each other. The alignment model can be based on $tanh$ (Bahdanau et al., 2015), cosine distance (Graves et al., 2014), dot-product, $softmax$, or with a trainable matrix (Luong et al., 2015).

Figure 3.7 shows the graphical illustration of attention model[4] which is generating y_t (the t^{th} word from output) from the source input x.

Nowadays most approaches use attention in one way or another. I highlight and list below some early works as they particularly related to the work I do.

The work of Hermann et al. (2015) (**The Attentive Reader**) presents a class of attention based deep neural networks for the reading comprehension task. They propose a generalization of the application of Memory Networks (Weston et al., 2015) (see next section) to question answering, where the attention model is used for encoding a document and a query by separate bidirectional single layer LSTMs. **Attention-over-Attention Reader** (Cui et al., 2017) (AoA Reader) puts another attention for answer prediction over

[4]The illustration is inspired by Bahdanau et al. (2015) and Lilian Weng `https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html` – last verified June 2021

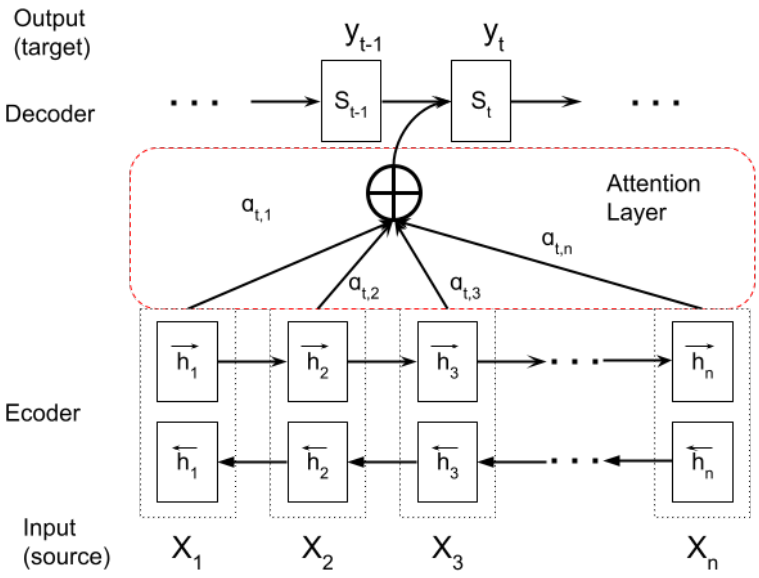


Figure 3.7: Illustration of attention architecture.

document-level attention. **Gated-Attention Readers** (Dhingra et al., 2017a) (GA Reader) uses a multi-hop architecture with an attention mechanism, which is based on multiplicative interactions between the embedding of the question and the intermediate states of a recurrent neural network document reader.

3.4.3 Memory Networks

In this section I briefly describe the concept of memory networks as I will describe an approach based on it in Section 4.2.3. Memory Networks are introduced by Weston et al. (2015) and use a model for inference combined with a long-term memory for prediction, which can be read and written to. An architecture consists of: *a memory*, which is an array of objects like vectors or strings; *input feature map* I, which is used for obtaining the internal feature representation; *generalization* G, which is responsible for updating the memory according to the new input; *output feature map* O, which is used to obtain the output features based on the new input and current memory state; and *response* R, which brings the output features into the final output. See Figure 3.8. Each of the four components I, G, O and R can have its own architecture, with each being a different type of neural network.

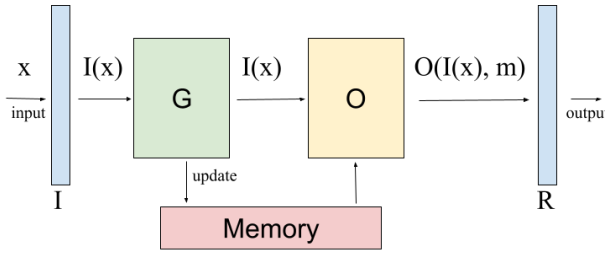


Figure 3.8: Illustration of Memory Network.

3.4.4 Transformer

The transformer as an architecture was introduced by Vaswani et al. (2017) and I use it for my experiments and analysis in Chapters 5-7. A transformer is based on the encoder-decoder paradigm but instead of traditional input which is read one element of the sequence at a time, it takes the entire input sequence at once. It uses N stacked fully connected layers[5] of two sub-layers (multi head self-attention and point-wise feedforward network) for the encoder and N fully connected layers of three sub-layers (multi-head attention over the output of the encoder stack, multi head self-attention and point-wise feed-forward network) for the decoder. It is the first model relying entirely on self-attention to compute representations of its input and output without using sequence aligned RNNs or convolutions. The illustration of the transformer architecture is presented in Figure 3.9.

To convert the input tokens and output tokens to vectors, the authors trained their model to learn embeddings. As there is no recurrence and no convolution, the relative or absolute position of the token are encoded into *position encoding* and added to the word embeddings and used to keep track of the order of the sequence.

3.4.4.1 Self-attention

Self-attention is an attention described above in section 3.4.2 with the following difference: while the attention is looking into matches between two different sequences (like the source and target sentences in an MT system), self-attention is considering as input and output the same sequence x. To learn the match of "attention" in the sequence there are three additional parameters: *query*, *key*, and *value*.

[5] In the original paper $N = 6$

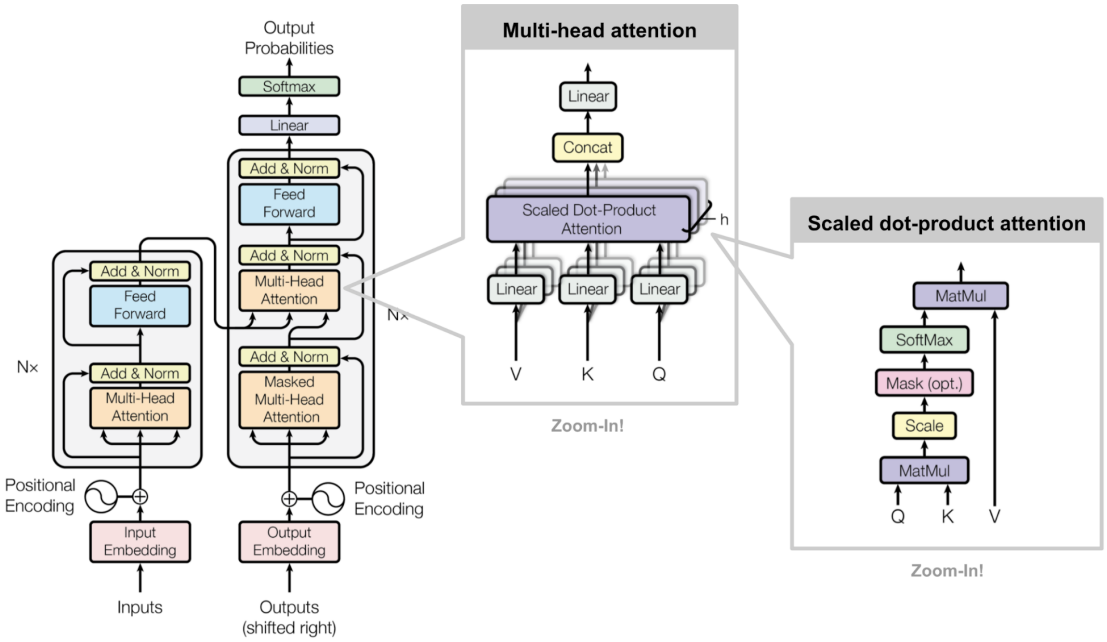


Figure 3.9: Illustration of Transformer architecture from Vaswani et al. (2017) combined by Lilian Weng: `https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html` – last verified November 2021.

- Query $q_i = W^Q x_i$ is the current focus of the attention and it is compared to all preceding inputs x_j;
- key $k_i = W^K x_i$ is an input being compared to the current focus of attention q_i;
- value $v_i = W^V x_i$ is used to compute the output for the current focus of attention.

Now the score between a current focus of attention x_i and an element in the context x_j can be calculated as a dot product of query and key of the context. Practically, to avoid large values it is scaled by square root from the dimension of keys d_k, see Equation (3.24):

$$score(x_i, x_j) = \frac{q_i \cdot k_j}{\sqrt{d_k}} \tag{3.24}$$

Then the output y_i is calculated according to Equation (3.25):

$$y_i = \sum_{j<=i} \alpha_{ij} v_j \tag{3.25}$$

where α_{ij} is a vector of weights, that indicates the relevance of each input element i to the current focus of attention j. It is calculated according to Equation 3.26:

$$\alpha_{ij} = softmax(score(x_i, x_j)) = \frac{e^{score(x_i,x_j)}}{\sum_{k=1}^{n} e^{score(x_i,x_k)}} \tag{3.26}$$

Generalizing over the entire input the individual vectors can be formed into matrix and take advantage of efficient matrix multiplication. So the entire self-attention can be calculated as shown on Equation (3.27):

$$SelfAttention(Q, K, V) = softmax(\frac{QK^\top}{\sqrt{d_k}})V \tag{3.27}$$

Figure 3.10 illustrates the use of self-attention in calculation of the third element of a target sequence.

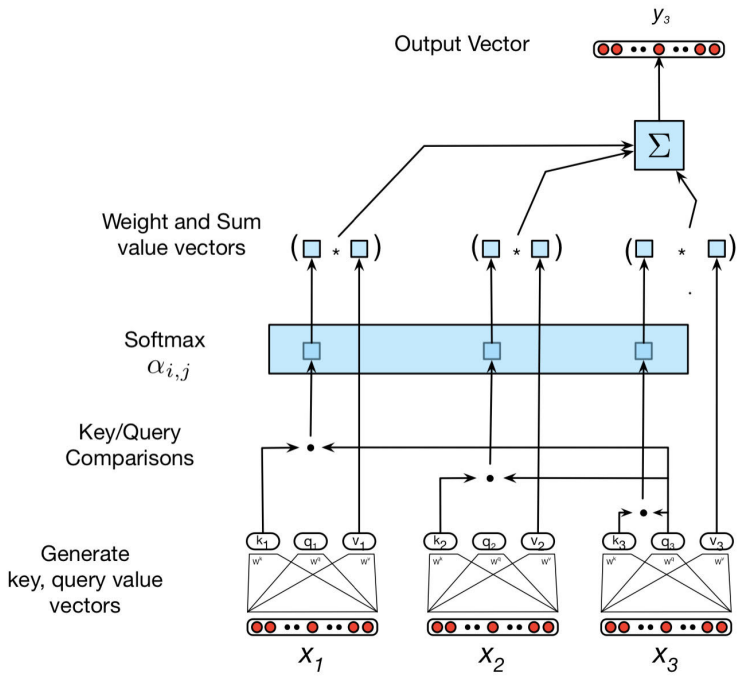


Figure 3.10: Illustration of self-attention from Jurafsky and Martin (2020).

Self-attention used in an encoder has access to the entire input while in a decoder it accesses only the preceding and current input, i.e. $j <= i$. This is critical for language modeling as predicting the next word should be learned without being known in advance.

3.4.4.2 Multi-head Attention

Vaswani et al. (2017) found it beneficial to use sets of self-attention layers (called *heads*) with different learned parameters h times. Each of these heads execute the attention

function in parallel with its own output which are concatenated together and brought back to the original output size by another learning parameter W^O as Equation (3.28) shows:

$$\begin{aligned} MultiHeadAttention(Q,K,V) &= W^O(head_1 \oplus head_1 \oplus ... \oplus head_h) \\ head_i &= SelfAttention(W_i^Q X, W_i^K X, W_i^V X) \end{aligned} \tag{3.28}$$

The advantage of multi-head attention is to provide the model with the ability to attend different positions of input at the same time and learn different ways how parts of the input can relate to each other.

3.4.4.3 OpenAI Transformer

Radford et al. (2018, 2019) introduces the OpenAI GPT transformer and show the advantage of using a transformer decoder for language modeling. The authors also show how a transformer architecture can be used for transfer learning by applying it to different types of tasks. Figure 3.11 from Radford et al. (2018) particularly illustrates input into transformer model and an added linear output layer for the text classification, entailment task, text similarity, and multiple choice for the question answering task.

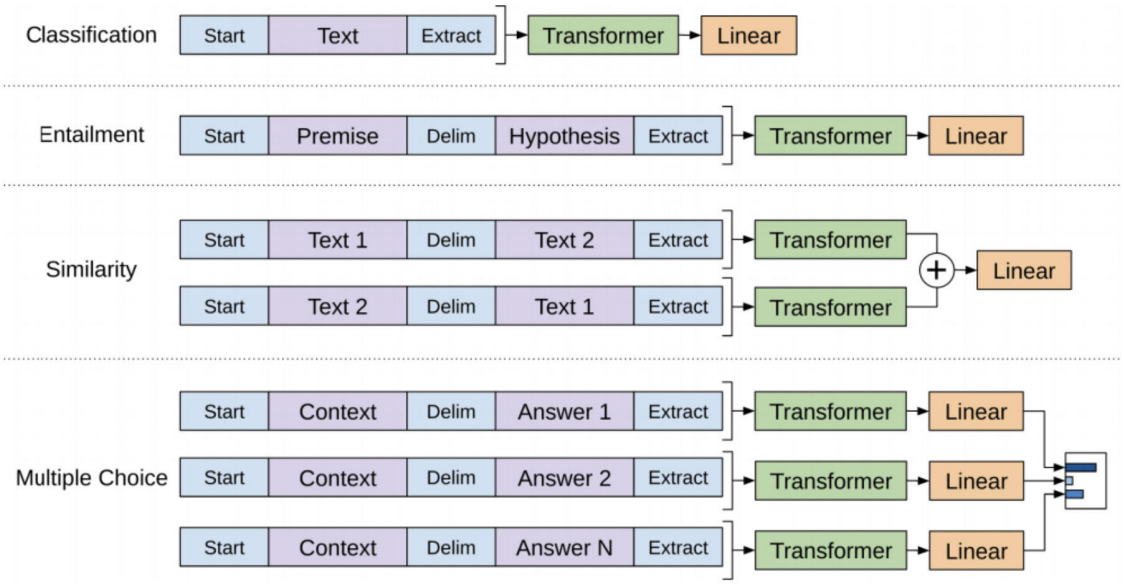


Figure 3.11: Illustration from Radford et al. (2018) of transformer usage for different type of tasks.

3.4.4.4 BERT

The Bidirectional Encoder Representations from Transformers (BERT) is a very popular and powerful transformer-based architecture proposed by Devlin et al. (2019). The works of Radford et al. (2018) (OpenAI GPT) learned only a forward language model which means to produce a token on j^{th} position it looks on all tokens of input $i : i <= j$. ELMO (Peters et al., 2018) is bi-directional which first uses a forward language model like OpenAI GPT, then it uses backwards language model trying to predict the current token based on the following part of input. In contrast, BERT uses transformer encoder with masked words for language model and next sentence prediction (NSP) task (See Figure 3.12).

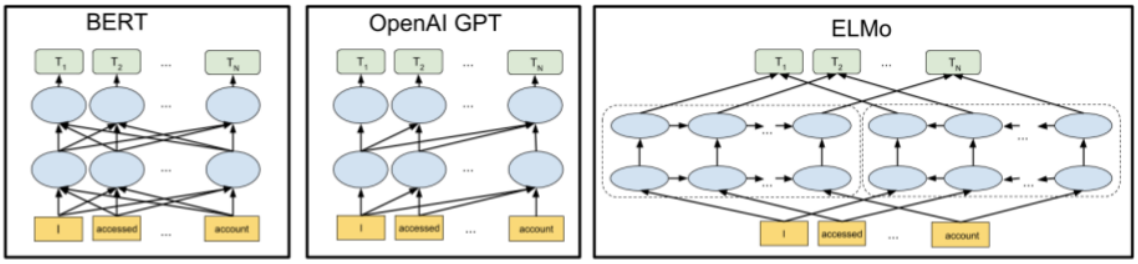


Figure 3.12: Difference between BERT (Devlin et al., 2019), OpenAI GPT (Radford et al., 2018), and ELMO (Peters et al., 2018). Illustration from Devlin et al. (2019).

Figure 3.13 presents the input embeddings for the BERT. The embeddings are the sum of the token embeddings, the segmentation embeddings (position before or after separation) and the position embeddings (indicate the token position in the sequence or input text), where [CLS] is artificially inserted *start* token and [SEP] is a separation token inserted in the end as well. The original input text is split into word pieces (WordPiece embeddings described by Wu et al. (2016) with a 30,000 token vocabulary is used) separating punctuation and word parts, e.g. “playing -> [’play’ ’##ing’]”, “attachments –> [’attachment’, ’##s’]”, “dishwasher –> [’dish’, ’##wash’, ’##er’]”. BERT has a limit on the length of an input sequence, restricting input to 512 word pieces including the start and separation tokens.

Figure 3.14 illustrates the usage of BERT for a number of NLP tasks such as the Sentence Pair Classification task (a), the Single Sentence classification task (b), the Reading

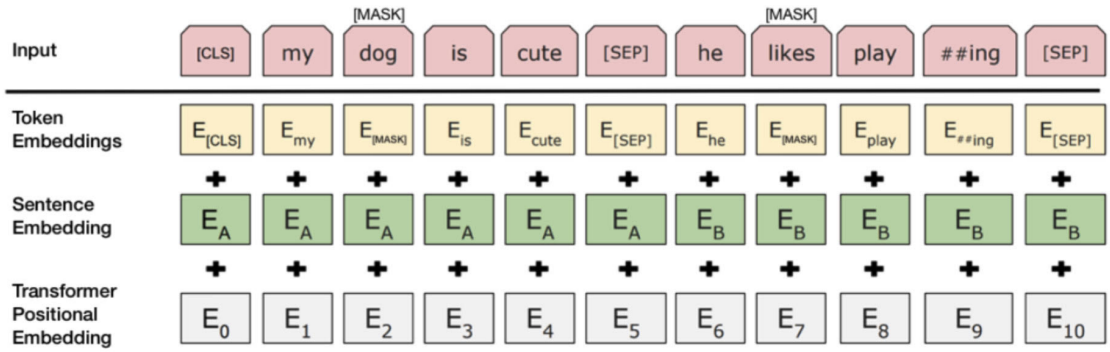


Figure 3.13: BERT input representation. Illustration from Devlin et al. (2019).

Comprehension (c) (SQuAD (Rajpurkar et al., 2016)), and the Single Sentence Tagging task (d).

There are two BERT models pretrained on BooksCorpus (Zhu et al., 2015) and Wikipedia available: $BERT_{base}$ (the encoder consists of 12 layers, hidden size is 768, and 12 self-attention heads) and $BERT_{large}$ (the encoder consists of 24 layers, hidden size is 1024, and 16 self-attention heads). There is a significant difference between time and memory usage between those two models during the fine-tuning process.

Since BERT has been created, there has been a lot of work done to understand how exactly it works. Rogers et al. (2020b) provide an overview of discovered facts about the BERT model, e.g. for the cloze task (masked word prediction) BERT pays attention to subject-predicate agreement but struggles with handling negations; BERT is also able to ”understand” some entity types, relations, and semantic roles but cannot reason on the learned word knowledge.

3.4.4.5 RoBERTa

Claiming BERT is undertrained, Liu et al. (2019b) present a replication of BERT called RoBERTa. They use the same model architecture and do careful exploration of key hyper parameters, training data size, and training on longer sequences. A performance improvement on the number of NLP tasks including reading comprehension question answering has been shown.

First of all, ten times more training data was used (160GB vs 16GB). In addition to

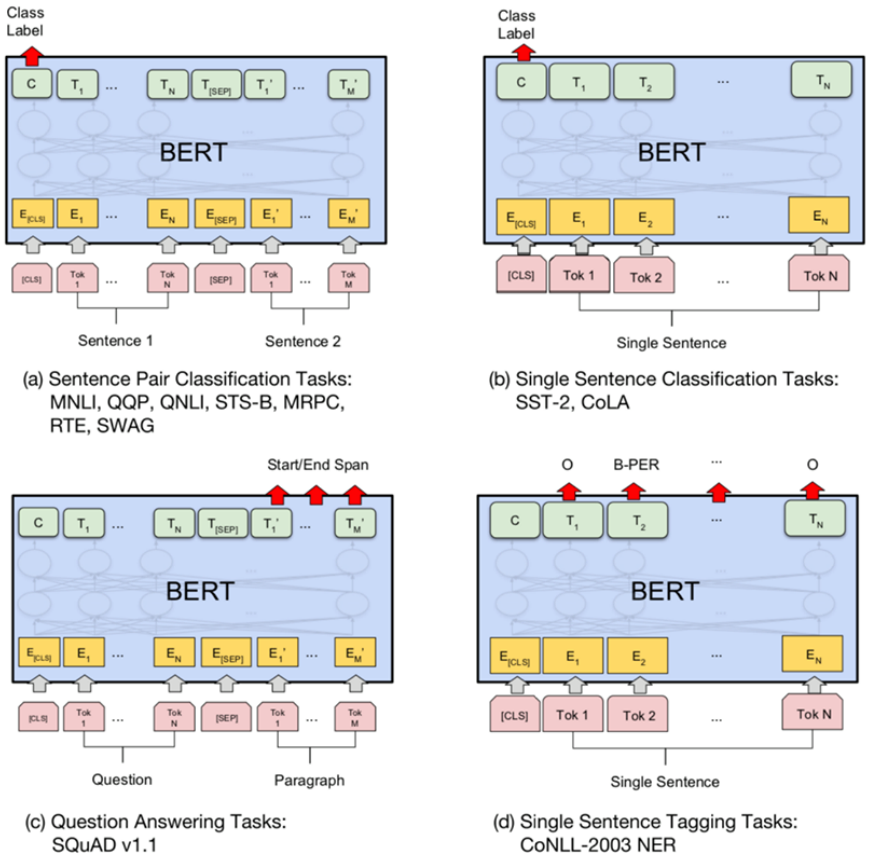


Figure 3.14: Applying BERT to a number of NLP tasks including Reading Comprehension. Illustration from Devlin et al. (2019).

BooksCorpus+Wikipedia, the CommonCrawl News dataset (Nagel, 2016), OpenWebText (Gokaslan and Cohen, 2019), and Stories dataset (Trinh and Le, 2018) were also used.

Additionally, the masking for language model is done differently. BERT masks training data once (*static* approach): in 80% of cases the word is replaced with the [MASK] token, in 10% cases it is replaced with a random word, and in 10% of cases the word remains unchanged. RoBERTa, on another hand, masks the data *dynamically*. The masking pattern is generated every time the sequence is fed to the model. It duplicates training data 10 times so that each sequence is masked in 10 different ways over the 40 epochs of training. That means, each training sequence was seen with the same mask only four times during training. In their experiments Liu et al. (2019b) call into question the usefulness of the next sentence prediction task.

3.4.4.6 SBERT

Pointing out that the averaged BERT output embedding or the output of the first $[CLS]$ token often is not as good as averaging GloVe embeddings, Reimers and Gurevych (2019) use two networks with the same weights (siamese network) on top of BERT (and RoBERTa) to obtain a fixed-sized sentence embedding. One of the settings they experiment with is regression where the cosine similarity between two sentences embedding is calculated, and mean-squared-error loss is considered as the objective functions (see Figure 3.15). The model is trained on the combination of the SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018). I use SBERT for sentence similarity in Chapter 7 .

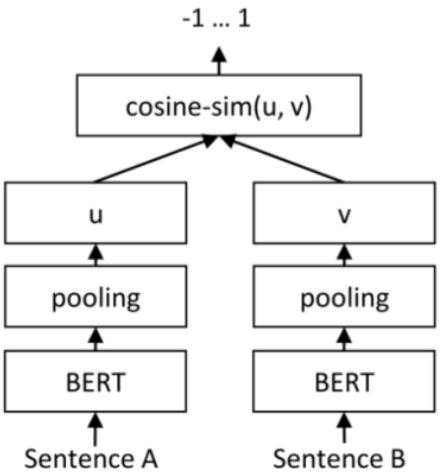


Figure 3.15: SBERT architecture at inference from Reimers and Gurevych (2019).

3.4.5 Graph Based Neural Networks

Graph Neural Networks (GNN) is another type of neural network. It was first proposed by Gori et al. (2005) and modified by Scarselli et al. (2009). It has seen considerable interest in recent years (Li et al., 2016; Kipf and Welling, 2017; Gilmer et al., 2017; Veličković et al., 2018). First I explain the general idea of GNN and then I provide details of GNN sub-layer extension for Transformer proposed by Shaw et al. (2019) as this is exact architecture I used in my experiments in Chapter 6.

3.4.5.1 Graph Neural Networks Fundamentals

GNNs are an extension of neural networks which work directly on graph-structured data. They are widely used for the classification task, as every node v in the graph G is associated with a label o_v which is defined by a s-dimensional vector h_v (state embedding), and the label of the nodes without a gold label should be predicted. See Equation (3.29).

$$\begin{aligned} h_v &= f(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]}), h_v \in \mathbb{R}^s \\ o_v &= g(h_v, x_v) \end{aligned} \tag{3.29}$$

where x_v are the features of the node v, $x_{co[v]}$ are the features of the edges connecting with the node v, $h_{ne[v]}$ is the embedding of the nodes in the neighborhood of v, $x_{ne[v]}$ the features of the nodes in the neighborhood of v (see Figure 3.16). Also, f is the local transition function and g is the local output function, which can be interpreted as feedforward neural networks.

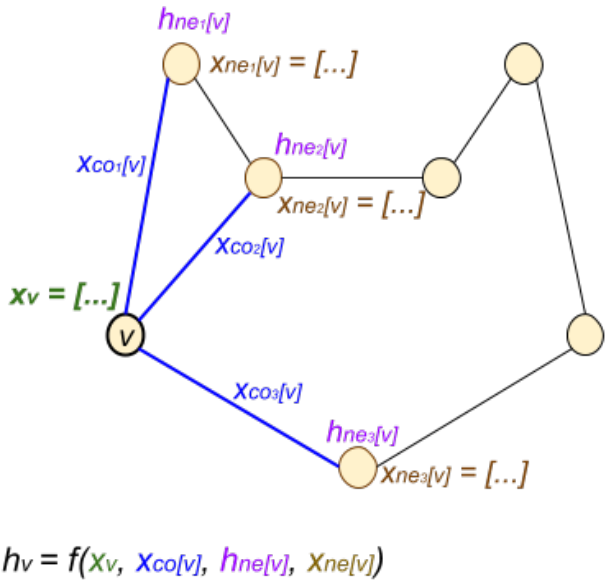


Figure 3.16: Illustration of GNN state embedding variables.

The Equation (3.30) generalise the Equation (3.29) with the global functions F and G.

$$\begin{aligned} &\forall v : H = \cup h_v; X = \cup x_v; X_N = \cup x_{ne[v]}; O = \cup o_v, \\ &H = F(H, X) \\ &O = G(H, X_N) \end{aligned} \tag{3.30}$$

Then the state H^{t+1} can be calculated using the following classic iterative scheme (3.31) from the t^{th} iteration based on Banach's fixed point theorem (Khamsi and Kirk, 2001):[6]

$$H^{t+1} = F(H^t, X) \tag{3.31}$$

Figure 3.17 provides an example of calculating the states for the particular node. At the first step ($t = 0$) the state contains information only about internal features of the node. On the next step the closest neighbors influence the state, etc.

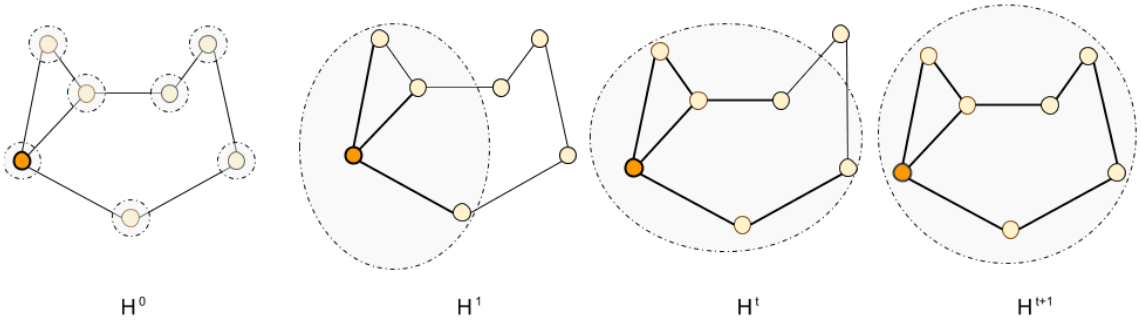


Figure 3.17: Illustration of iterative calculation of a graph state embedding H for the node v on each step t.

According to Zhou et al. (2018), the loss-function for supervised setting can be defined as Equation (3.32):

$$loss = \sum_{i=1}^{p} (r_{v_i} - o_{v_i}) \tag{3.32}$$

where r_{v_i} is a target information for specific node v_i, $p = |V_{train}|$ is the number of nodes available for training.

More details of methods and applications of GNN can be found in the works of Zhou et al. (2018); Hamilton (2020); Li and Saúde (2020).

[6]Banach–Caccioppoli fixed-point theorem guarantees the existence and uniqueness of fixed points of certain self-maps of metric spaces.

3.4.5.2 GNN for Transformer

Now let's look in the GNN sub-layer for the transformer.

Let f be a function over a node representation m and edge label l. Shaw et al. (2019) use edge labels representation as additive vectors (see (3.33)):

$$\begin{aligned} f &: \mathbb{R}^d \longrightarrow \mathbb{R}^{d'} \\ f(m, l) &= W^r m + w^l \end{aligned} \tag{3.33}$$

where $d' = d/n_{heads}$, n_{heads} is a number of used parallel attention heads, $W^r \in \mathbb{R}^{d' \times d}$ is a learned matrix shared by all edge labels, $w^l \in \mathbb{R}^d$ is a learned embedding vector per edge label l.

If f is implemented as $f(m, l) = W^r m$, then the sub-layer would be the same as self-attention in the Transformer.

The GNN sub-layer is used to map an ordered sequence of node representations $\mathbf{u} = (u_1, \ldots, u_n)$ from a fully connected, directed graph with edge labels r (r_{ij} is the edge label between u_i and u_j) into a new sequence of node representations **u'** $= (u'_1, \ldots, u'_{|\mathbf{u}|})$:

$$\begin{aligned} u_i^{k\prime} &= \textstyle\sum_{j=1}^{|\mathbf{u}|} \alpha_{ij} f(u_i, r_{ij}) \\ \alpha_{ij} &= softmax(\tfrac{(\mathbf{W}^q u_i)^\intercal f(u_i, r_{ij})}{\sqrt{d'}}) \\ u'_i &= \mathbf{W}^h [u_i^{1\prime}, \ldots, u_i^{n'_{heads}}] \end{aligned} \tag{3.34}$$

where α_{ij} is a softmax over the scaled dot products (as 3.34 shows), $\mathbf{W}^q$ and $\mathbf{W}^h$ are matrices to learn, [...] denotes concatenation.

3.5 Summary

In this chapter I have summarised the basics of the approaches widely used in the reading comprehension task. I mostly paid attention to those architectures which are needed to describe the experiments presented in this thesis and comparable methods.

Chapter 4

Multiple Choice Question Answering

> Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better.
>
> *Edsger Wybe Dijkstra*

In this chapter I address the second research question which is related to the multiple choice reading comprehension task. This work has been done in 2016-2017 before the majority of multiple choice datasets reviewed in Chapter 2 were released. I propose a data-driven similarity-based approach for answer selection. Back then, the system I developed achieved the highest accuracy on the Plot Synopses subtask for the MovieQA dataset introduced by Tapaswi et al. (2016). Another version of my system won the IJCNLP Shared Task Nº 5 Multi-choice Question Answering in Examinations (MCQA) (Shangmin et al., 2017).

The chapter is structured as follows: I start with a description of the proposed approach based on string similarities and logistic regression in Section 4.1. Section 4.2 contains the related work, experimental setup, and results for the Wikipedia-based dataset MovieQA. In Section 4.3 I describe the related work, experimental setup, and results for the MCQA Shared Task based on examination questions. Finally, I discuss the results and

summarize the chapter in Section 4.4.

4.1 Logistic Regression over String Similarity

I follow a natural reading comprehension strategy: consider the question first to find the answer in the text; then, based on the found evidence, select the most suitable answer.

The proposed approach is organized as a pipeline. The first (optional) step is text pre-possessing. The second step is an evidence search which I do by SENTENCE SELECTION. The main idea of sentence selection is to focus on finding content that is directly relevant to a question. The third step is CALCULATION OF SIMILARITIES, and, finally, ANSWER SELECTION with logistic regression.[1]

4.1.1 Sentence Selection

There are several different ways to extract relevant information from text. The first one is based on a number of similarities described in Section 3.2 and extracts **individual sentences**. Using those similarities, n sentences from a passage related to the question are extracted. Subsequently, the extracted sentences are concatenated into one string. Duplicate sentences are removed.

Formally: considering the passage as a set of sentences P, question q, and a set of answers A_q, F is a set of features derived from a number of similarity methods Φ (see Section 3.2). First, n relevant sentences $S^n_\phi \subseteq P$ are extracted by applying ϕ, where $\phi \in \Phi$ as in (4.1), and n is the number of sentences to be selected.

$$\forall j \phi_j \in \Phi : S^n_{\phi_j} = \{s_1, ...s_n : max_n(\phi_j(q, s_i)), \forall i \; s_i \in P\} \tag{4.1}$$

All sentences from $S^n_{\phi_j}$ are concatenated to one string s_n, as in (4.2), where n is fixed and $m = |\Phi|$. This method concatenates sentences together in order of their relevance to the question.

[1] At the early stage of these experiments, I also tried other approaches such as Naive Bayes, SVM, and Decision Trees, but Logistic Regression showed better performance. In the following experiments, I focused on hyper-parameter tuning and an exploration of the feature space.

$$s_n = s_{1,\phi_1} \oplus \cdots \oplus s_{n,\phi_1} \oplus \cdots \oplus s_{1,\phi_m} \oplus \cdots \oplus s_{n,\phi_m} \tag{4.2}$$

The second way to extract relative information is to select a window of several sentences and pick up the most relevant **contiguous part** of the text. In other words, it is the same approach as described before but instead of selecting n individual sentences and concatenating them together to s_n, a set of n adjacent sentences $\bar{s_n}$ is selected, as shown in (4.3):

$$\forall \phi \in \Phi : \bar{s_n} = s_i, \oplus \cdots \oplus s_{i+n-1} : max_i(\phi(q, \bar{s_n})) \tag{4.3}$$

The third method is based only on a **sentence-level** similarity. The idea of adding sentence-level similarities was inspired by the results published on the MovieQA leaderboard by University College London[2] (see Section 4.2.3). For every question, the passage P is extended with question q and answer candidates $a_j \in A_q$. This extension is necessary to create the full term vocabulary, including those words which are present in the query or/and answer candidates only. All those sentences form a collection of documents where each sentence is considered as a document in terms of `TF-IDF`. Cosine similarity of `TF-IDF` representations is calculated between every sentence in the passage and the question, and between every sentence in the passage and every answer candidate for the question. As shown in (4.4), the result is the maximum of the sum of these two similarities over all sentences and all answers. This way of selecting sentences not only selects a sentence but also gives the general question-answer-sentence similarity feature f'. Although I use `TF-IDF`, any representation t can be applied:

$$\begin{aligned} & f'_j = max_{s_i \in P}(cos(t(q), t(s_i)) + cos(t(a_j), t(s_i))), \\ & \forall j \; a_j \in A_q \end{aligned} \tag{4.4}$$

[2]Using the term *sentence-level* I follow a terminology of method description from the leader-board: `http://movieqa.cs.toronto.edu/leaderboard/#table-plot`.
Technically, the first approach is also operating on the sentence-level but only for sentence selection. The feature similarities are calculated over a number of sentences, while in the third approach the feature similarities are calculated on a sentence-level.

4.1.2 Question Answer Concatenations

Previously, in Section 3.2 I discussed a number of ways to calculate the similarities between two strings. One of those strings is selected from the supported text sentences which form a passage (Str_1). How I obtain those sentences was just explained in Section 4.1.1. In this section I talk about the second argument of the similarity function, Str_2.

$$Str_2 = C(q, a) = \begin{cases} a, & \text{just an answer} \\ a + q, & \text{concatenation of answer and question} \\ q_s + a + q_e, & \text{insert answer into statement question} \end{cases} \tag{4.5}$$

It make sense to calculate the similarity between selected sentences and an answer option a as the sentence is already selected based on the question. Also, the selected sentences can be compared with the concatenation C of the question and answer candidate, although for cloze-style questions, the question-answer concatenation could be done in a better way by reconstructing the original sentence. So equation (4.5) shows all three options for the second string where $q_s + a + q_e$ is produced by filling the gap in the question with the answer candidate (cloze-style questions), where q_s is the start of the question-sentence before the gap, and q_e is the end of question-sentence after the gap.

4.1.3 Similarity Calculation

Once a number of sentences selected for every question are obtained, the similarity described in Section 3.2 can be calculated. The first argument of the similarity function can be one sentence s_i from passage P or several relevant sentences s extracted as described above and concatenated together. The second argument is a combination $C(a, q)$ of answer candidate and question, where C is a combination function. It can be just an answer, a simple concatenation, or gap filling as discussed in Section 4.1.2. The final set

of similarity features $V = \cup_n(V_n \cup \bar{V_n}) \cup V_{\sum n} \cup V'$ can be described by 4.6:

$$\begin{aligned} &\forall \psi \in \Phi : \\ &V_n = \{f : f = \psi(C(q,a), s_n)\}, \\ &\bar{V_n} = \{f : f = \psi(C(q,a), \bar{s_n}), \} \\ &V' = (f'_1, ..., f'_n) \end{aligned} \tag{4.6}$$

where $\psi \in \Phi$ is a similarity,[3] V_n are features which are calculated with n selected sentences, $\bar{V_n}$ are features with calculated with n contiguous sentences, $V_{\sum n}$ is an element wise sum of feature vectors across different n, and V' is a set of sentence-level similarity features.

Motivated by the idea that one sentence might have not enough information to answer the question and n sentences might be too noisy, the answer would lie between those two vector representations in a vector space (see Figure 4.1). Then the cosine similarity with the sum of those vectors might be closer to the answer than the original vectors. The sum of the vectors is an optional additional component.

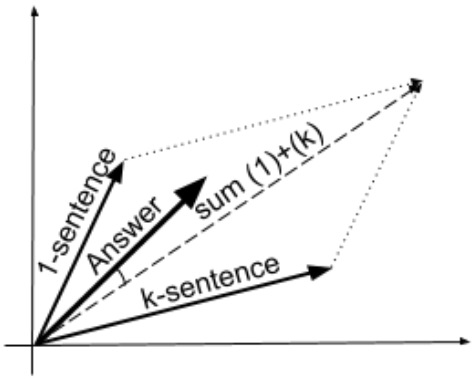


Figure 4.1: Motivation of usage sum of sentence vectors.

4.1.4 Vector Concatenation

The similarities are concatenated into one vector, meaning vectors for a number of n sentences are concatenated together and then concatenated with an element-wise sum of

[3]Previously I mark similarity which was used for sentence extraction as ϕ, now the similarity which is used as features I denote ψ. Both ϕ and ψ are from the same set of similarity metrics Φ.

those vectors.

Finally, the feature vector is extended with sentence-level similarities ψ for every answer and also with similarities between answers and contiguously extracted sentences. The full concatenation process is presented in Figure 4.2.

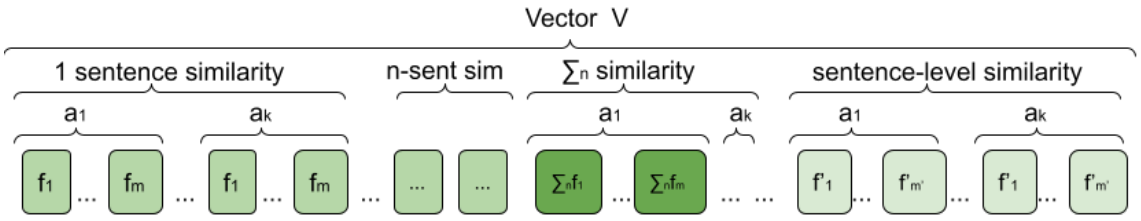


Figure 4.2: Feature vector concatenation, where m is the number of similarities and m' is the number of sentence-level similarities.

A high-level overview of the sentence selection process and similarity calculation is presented in Figure 4.3

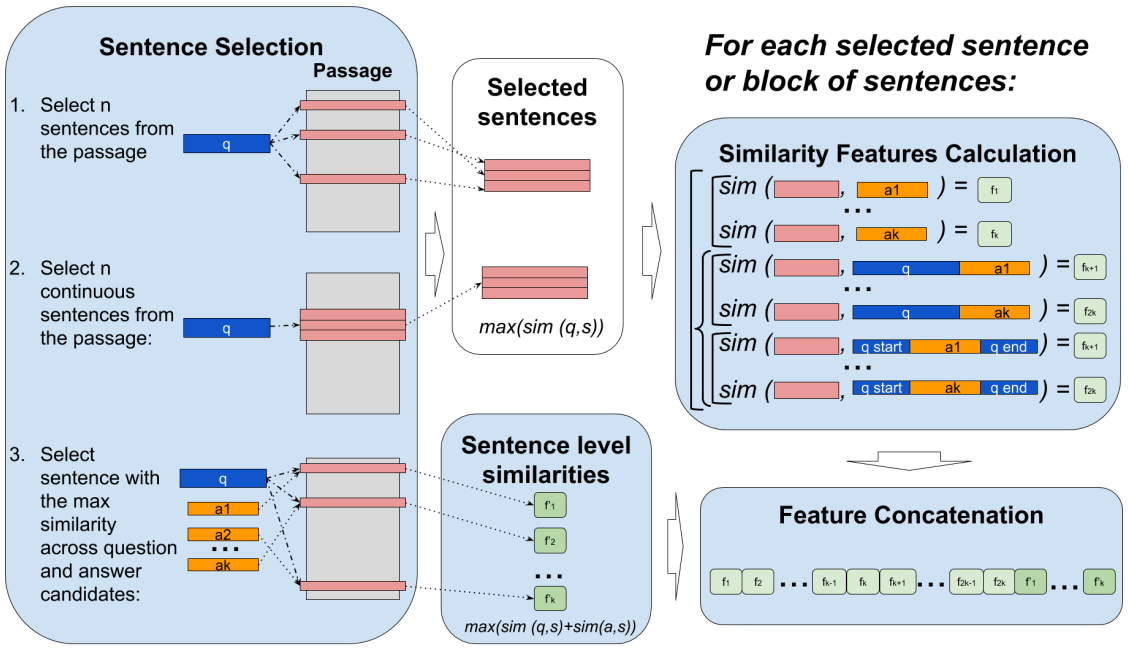


Figure 4.3: The high-level overview of the proposed approach: left block and bottom middle – $n = 3$ sentence selection, contiguous sentence selection, and sentence-level features calculation (see description in 4.1.1); top right block – question answer concatenation and similarities calculation (see description in 4.1.2 and 4.1.3); bottom right block – feature concatenation (see description in 4.1.4).

Above, I described the feature vector based on different text similarity measures. The vector contains the information about a question and all its answer candidates. In the final

stage, a logistic regression which predicts the answer is applied. Specifically, the idea is to detect the most relevant part of the feature vector and obtain the answer. In other words, my goal is to find out which similarities better represent matching between question, text passage, and answers.

4.1.5 Applying Logistic Regression

I consider answer selection as a classification task, where the logistic regression learns the labels of answers, and the labels are: *"First answer"*, *"Second answer"*, *"Third answer"*, etc. Vectors for each answer candidate are concatenated to one vector V, as described above:

$$V = [f_1^1, f_2^1, ...f_m^1, f_1^2, ...f_m^2, f_1^k, ...f_m^k,] = [f_1, f_2, ...f_m, f_{m+1}, ...f_{2*m}, f_{2*m+1}, ...f_{k*m}] \tag{4.7}$$

where m is the number of similarity features, and k is the number of answer candidates. For simplicity I omit the details of features here and just denote it f.

The vector V in Equation (4.7) for logistic regression is a concatenation of different similarities across the answers. So logistic regression is trying to learn which features and which part of this vector is the most important. And correspondingly, the particular part of the vector represents a particular class: the first m features are the first answer candidate (*class 1*), m+1 – 2m are the second (*class 2*) etc.

4.2 Answering MovieQA Multiple Choice Questions

As mentioned above in Chapter 2, the majority of multiple choice datasets were released after 2017.[4] Focusing on the multiple choice reading comprehension task I selected MovieQA as my main dataset.

[4]The RACE dataset was released only in September 2017.

4.2.1 MovieQA Dataset

The **MovieQA**[5] is a multi-modal dataset introduced by Tapaswi et al. (2016). It contains 14944 multiple choice question answers obtained from Wikipedia plot synopses of 408 movies. Apart from plot synopses, it also contains additional resources: video+subtitles, subtitles, the Descriptive Video Service (DVS) provided by Rohrbach et al. (2015), scripts, and open ended settings (any additional source or combination of sources can be used). Every question is accompanied by three to five answer candidates, with only one correct answer. The task is to select the correct answers using additional resources. Every question is annotated with a movie from Internet Movie Database (IMDb).[6]

(4.8) *Who kills Sirius?*

Movie: "Harry Potter and the Half-Blood Prince", 2009, (IMDB ID: *tt0417741*)

(4.9) *When was Boris captured?*

Movie: "Men in Black 3", 2013 (IMDB ID: *tt1409024*)

(4.10) *Does Batman manage to escape from the prison?*

Movie: "The Dark Knight Rises", 2012 (IMDB ID: *tt1345836*)

(4.11) *What are Jack's attempts to save Rose after Titanic's sinking?*

Movie: "Titanic", 1997 (IMDB ID: *tt0120338*)

(4.12) *How does Marian feel about Robin's band?*

Movie: "Robin Hood: Prince of Thieves", 1991 (IMDB ID: *tt0102798*)

(4.13) *Why does Octavius kidnap Mary Jane?*

Movie: "Spider-Man 2", 2004 (IMDB ID: *tt0316654*)

The dataset contains factoid, e.g. (4.8)-(4.10), and non-factoid e.g. (4.11)-(4.13) questions. Table 4.1 contains examples of factoid and non-factoid questions with answer candidates. I focus on the movie plots setting where there is a Wikipedia based movie plot

[5] `http://movieqa.cs.toronto.edu/home/` – last verified May 2021
[6] `http://www.imdb.com/` – last verified May 2021

available. The dataset is split into training, test and development sets as shown in Table 4.2.

Title: '71, 2014

Plot synopsis: Gary Hook, a new recruit to the British Army, takes leave of his much younger brother **Darren**. ... Hook steps outside the pub just before an enormous explosion destroys the building. **Hook flees once more into the dark streets.** ...

Factoid question:	**Non-factoid question:**
What is the name of Hook's younger brother?	How does Hook react to the explosion?
– His name is Carl	– He flees into the building next door
– **His name is Darren**	– He goes back into the pub to check for survivors and help the wounded
– His name is Jimmy	– He finds a payphone and calls the police
– His name is David	– **He flees into the street**
– His name is Tom	– He yells for help

Table 4.1: Example of factoid and non-factoid questions and candidate answers from the MovieQA dataset. Bold marks the relevant parts of the plot synopsis and the correct answer.

	Train	**Val**	**Test**
#Movies	269	56	83
#QA	9848	1958	3138

Table 4.2: Number of instances in the training, development, and test sets of the MovieQA dataset.

4.2.2 Experiments

I applied the exact pipeline described in Section 4.1 and depicted in Figure 4.3. The lengths of plot synopses vary widely, ranging from one to 20 paragraphs. To cut down execution time, sentence-selection is done with a limited number of similarities: `TF-IDF`, `Window slide`, `Bag of words`, and `Character N-gram` . Then the set of similarities extended with `word2vec` and `SkipThought` was applied to (i) selected sentences and answers, and (ii) selected sentences and concatenation of question and answer. I use $n = 1, 3, 5$.

For `Word2Vec` a pre-trained model based on a small[7] subset of Wikipedia provided by Tapaswi et al. (2016) was used. To calculate SkipThoughts-based similarity the question, answers, story and selected sentences were encoded with the pre-trained model (Kiros et al., 2015).

4.2.2.1 Results

Table 4.3 contains the results for logistic regression over different combinations of features. `1(+)3(+)5` is a concatenation of vector similarity for one extracted sentence, three extracted sentences and five extracted sentences. `(1+3+5)` is an element-wise sum of the described vectors. All four components together are concatenated to one vector: `1(+)3(+)5(+)(1+3+5)` (Table 4.3, line 1). The addition of `SkipThought` features, which were described in Section 3.1.2, (Table 4.3, line 2) leads to a slight degradation on the development set, so it can be concluded that the `SkipThought` representations do not contribute anything for my method. As described before, the vector is extended by adding `TF-IDF` similarity on the sentence level for every answer option. The performance of this combination on the development set is 78.29% (Table 4.3, line 3). The system is run with `TF-IDF` on sentence-level but without `SkipThought` similarity and obtained the best result on the development set – 78.39% and on the test set as well – 79.76% accuracy. The addition of similarities based on the contiguous extraction of sentences from plot, which were described in Section 4.1.1, improves this result on the development set (80.13%) and on the test set (80.02%). This result outperforms the previous state-of-the-art at that time (October 2017) accuracy of 79.99% obtained by Attention-based CNN Matching Net (Liu et al., 2017).

4.2.2.2 Feature performance

Additionally, I evaluate the performance of each feature separately (see Table 4.4). This is done by using only one feature at a time as input. The sentence-level `TF-IDF` similarity

[7]Tapaswi et al. (2016) use the skip-gram model to train domain-specific 300-dimensional word2vec representation on 1200 movie plots.

Feature combination	**Dev**	**Test**
1(+)3(+)5(+)(1+3+5)	74.8	76.04
+ SkipThought	74.36	-
+ SkipThought + S-level TF-IDF	78.29	-
+ S-level TF-IDF	78.39	79.76
+ S-level TF-IDF + ContiguousExtraction	**80.13**	**80.02**

Table 4.3: Performance on development and test sets, where *1(+)3(+)5(+)(1+3+5)* is concatenation of vector similarity for one extracted sentence, 3 extracted sentences and 5 extracted sentences; the last component is the element-wise sum of the described vectors.

achieves an accuracy of 72.52% on the development set. The majority of string similarity features works within a range of 50-63% accuracy. Notably, SkipThought features perform within a range of 25-48% accuracy. The low output of SkipThought features explains why excluding this representation improves the overall result of the model.

One of the reason why Word2Vec features was not effective could be the size of the model. I use a model which was pretrained by Tapaswi et al. (2016) on a subset of 1200 Wikipedia articles about movies. A bigger model might provide better results.

	1 sentence		3 sentence		5 sentence	
	A vs S	qA vs S	A vs S	qA vs S	A vs S	qA vs S
w2v basline	47.75		47.34		47.24	
w2v cos	43.36	48.51	29.67	28.44	27.78	25.12
SkipThought cos	34.98	32.99	33.65	27.68	33.75	26.40
TF-IDF	59.65	62.25	53.67	56.12	50.61	53.26
BOW Overlap	60.92	61.49	61.18	62.81	59.60	61.64
WindowSlide	61.64	45.14	64.19	52.14	63.89	53.21
2-gram	55.00	56.12	42.39	46.37	36.51	40.04
3-gram	62.46	63.17	60.77	62.20	57.55	59.95
4-gram	61.95	62.05	64.35	64.65	63.99	64.65
5-gram	59.95	60.62	62.61	63.78	62.76	64.19
S-level TF-IDF	72.52					

Table 4.4: Separate performance of all features on the development data (accuracy). Where *A vs S* is a similarity between answer and sentences. *qA vs S* is a similarity between a concatenation of question and answer, and sentences.

The results of running logistic regression over only one, three, five selected sentences, and the sum of them are presented in Table 4.5. I observe the accuracy increases when increasing the number of extracted sentences. As was shown before, the best result is

obtained by a combination of the features.

	1 sentence	3 sentence	5 sentence	Sum
Logistic Regression	61.64	71.34	72.36	74.36
Logistic Regression +S lvl TF-IDF	76.86	77.22	77.17	78.19

Table 4.5: Results of logistic regression on the development set over separate feature combinations for one, three, five extracted sentences and also the sum of them.

4.2.2.3 Error analysis

In this section, some errors in the development set are discussed. I inspect 85 samples from the development set which is around 22% of all errors. Three main classes of errors can be highlighted: *Misunderstanding*, *Incorrect/No aggregation*, *Incorrect sentence selection*. Note that in many cases these classes are overlapping.

Around 70% of errors are caused by misunderstanding the context. This category involves alternative phrasing, including synonyms, and wrong references. For example, in (4.14) the system probably did not make a connection between the "*profession of dead husband*" and "*widow of*", and answering the question in (4.15) requires a resolution of references: *Walter* is also called *Neff* and "*drives*" means "*on the ride*".

(4.14) **Movie:** "Broken Flowers", 2005 (IMDB ID:*tt0412019*)

Question: What was Laura's dead husband profession?

Plot Synopses: Laura (Sharon Stone) works as a closet and drawer organizer and is *the widow of a race car driver. ...*

Predicted Answer: Closet and drawer organizer

Correct Answer: Race Car Driver

Explanation: Rephrasing: *widow of — dead husband profession*

(4.15) **Movie:** Double Indemnity, 1944 (IMDB ID:*tt0036775*)

Question: Where does Walter kill Dietrichson?

Plot Synopses: *Walter Neff* (Fred MacMurray), ... After *Dietrichson* breaks his leg, *Phyllis drives him to the train station* for his trip to Palo Alto for a college

reunion. *Neff* is hiding in the backseat and *kills Dietrichson* when Phyllis turns onto a deserted side street. ...

Predicted Answer: At Phyllis' house

Correct Answer: On the ride to the train station

Explanation: Information across the sentences.

References: *Walter — Neff*.

Rephrasing: *drives — on the ride*

To capture semantically similar words, the method includes semantic representation (`Word2Vec` and `SkipThought`). Apparently, as discussed in Section 4.2.2.2, my use of these features does not work satisfactorily.

The second large class of error (about 32.5%) is based on the fact that some questions request information which is spread across two or more sentences, e.g. Examples (4.15)–(4.16). The idea to select more than one sentence (three and five sentences as well) came from a desire to increase the probability of selecting the right sentence and also to be able to analyze information across the sentences. In the current implementation, the sentence selection is working on a sentence level. Also, the order of sentences is ignored during the selection of 1-3-5 sentences as the most relevant text comes first and might change the order of the original sentences. For such similarities as BOW Overlap, the order of the sentences (or even words) does not matter, but WindowSlide is sensitive to the sentence order and reordered sentences can confuse the model.

(4.16) **Movie:** "Confessions of a Shopaholic", 2009 (IMDB ID:*tt4440352*)

Question: What is the name of the leader of the Shopaholics Anonymous group?

Plot Synopses: Rebecca later returns home to renewed confrontations with her debt collector, so Suze makes her attend *Shopaholics Anonymous*. ... After one shopping spree she meets a friendly woman, *Miss Korch* (Wendie Malick), only to learn that she *is the group leader* and ... .

Predicted Answer: Suze

Correct Answer: Miss Korch

Explanation: Information across the sentences.

Only around 8% of questions were supported by sentences which do not contain the correct answer information. See Example (B.1) in Appendix for such question and extracted sentences.

4.2.3 Related work

By the end of the year 2017, my result was on the top of the leader-board with the best result of 80.02% accuracy on the Plot Synopses setting task. The result remained at the top for 5 months. The results for the MovieQA plot dataset including this work are presented in Table 4.6 excluding anonymous submissions and similar submissions from the same teams.

Tapaswi et al. (2016) provide four baselines for finding the correct answer:
(1) *Hasty Student* (not in the table) chooses answers without looking into additional text. The best result was 28.14% accuracy on a question-answer similarity of sentence-level SkipThoughts Vectors (see Section 3.1).
(2) *Searching Student (SS)* (not in the table) selects the answer based on a cosine similarity between `TF-IDF`, `SkipThoughts`, and `Word2Vec` representations of question-answers and corresponding additional data sources (Wikipedia movie plots in this case).
(3) *Searching Student with Convolutional Brain (SSCB)* is a neural similarity model which considers the same representation as SS and also combinations. Empirical evaluations show that SSCB is sensitive to initialization. The result of different runs shows differences of up to 30% accuracy. The authors trained several networks using random start and picked the model with the best performance on the internal development set. This method achieves accuracy of 57.97% on plot synopsis data (Table 4.6, lines 5 and 6).
(4) *MemN2N* is a memory based approach proposed by Sukhbaatar et al. (2015) (see Section 3.4.3 for a description of memory networks). Tapaswi et al. (2016) modified the architecture with an additional embedding layer which encodes each multi-choice answer and uses an attention mechanism to find a relevant part of the story to the question. It

achieves 38.43% accuracy on the test set (Table 4.6, line 14).

Wang and Jiang (2017) propose a four layer LSTM model and investigate different comparison functions. This system achieved an accuracy of 72.9% on the test set and 72.2% on the development set (line 11) using a combination of operations: `SUBTRACTION`, `MULTIPLICATION`, and `NEURALNET` (ReLU).

The usage of sentence-level `TF-IDF` similarity, which was discussed in Section 3.2, was originally proposed by the Machine Reading Group[8] from University College London. This method shows 75.78% accuracy on the test set (Table 4.6 line 10). Another result from the same team is 77.63% on the test set (Table 4.6, line 9). They used the SSCB method described by Tapaswi et al. (2016) with sentence-level `TF-IDF` approach and `Word2Vec` representation. Unfortunately, no article is provided.

Liu et al. (2017) achieved accuracy of 79.99% (Table 4.6, line 7). They represent paragraphs of the plot, a question, and answer options as a tensor and use a sophisticated attention method to integrate them. A model consists of 3 layers: the first is a similarity mapping method, which computes the word embedding similarity between every word in the paragraph and answer option (or question); the second is the attention based CNN matching; the third is a prediction layer which determines the final answer.[9]

By September 2020 a few more results have been recorded. The best one is 87.79% accuracy obtained by Mossad et al. (2020). They call the architecture FAT ALBERT. First they extract 5 sentences from the movie plot and then apply BERT for multiple choice question answering. To extract relevant sentences they first use BERT to obtain embeddings for two sentences, then calculate a number of similarities (cosine, n-gram,[10] and levenshtein distance), and then use a fully connected neural network with softmax layer to obtain the similarity index.

The next few results are provided by Blohm et al. (2018) who based their approach on a combination of the two-stage attention mechanism with the general compare-aggregate architecture proposed by Wang and Jiang (2017), which is a four level model consisting

[8] `http://mr.cs.ucl.ac.uk/` – last verified May 2021

[9] The description of the system is taken from the MovieQA leader-board.

[10] Mossad et al. (2020) use a term q-gram, which is the same according to Ukkonen (1993).

of a preprocessing layer, attention, comparison layer, and aggregation (CNN) for the final classification prediction. Blohm et al. (2018) used the aggregation function separately on a word and sentence level, and experimented with RNN and CNN implementations of the aggregation function.

System	Date	Val	Test
FAT ALBERT (Mossad et al., 2020)	10 Dec 2019	**87.28**	**87.79**
Attention-Based Matching Network (LSTM) (Blohm et al., 2018)	04 Aug 2018	84.37	85.12
Attention-Based Matching Network (CNN+LSTM) (Blohm et al., 2018)	29 May 2018	84.78	84.70
CNN Matching Network with Sentence Attention (Blohm et al., 2018)	14 Mar 2018	82.58	82.73
RangeR encoder + Attention and Aligment	08 Mar 2018	-	80.72*
Logistic Regression (Dzendzik et al., 2017b)	13 Oct 2017	80.13	80.02
Attention based CNN Net (Liu et al., 2017)	14 Sep 2017	79.00	79.99
Tensor representation	06 Aug 2017	-	78.52*
Convnet (TF-IDF + w2v)	26 Jan 2017	-	77.63*
TF-IDF on sentence-level	17 Nov 2016	72.73	75.78*
CNN on word matching (Wang and Jiang, 2017)	18 Sep 2016	72.10	72.90
SSCB tfidf + w2v (Tapaswi et al., 2016)	02 Jan 2017	59.60	57.97
SSCB Fusion (Tapaswi et al., 2016)	-	61.24	56.70
MemN2N (Tapaswi et al., 2016)	24 Oct 2016	40.45	38.43

Table 4.6: The state-of-the-art results for the MovieQA plot dataset. * - results are obtained from the MovieQA Leaderboard.

4.3 Answering Multiple Choice Questions from Examinations

Subsequent to my work on the MovieQA dataset I participated in the IJCNLP Shared Task with the MCQA dataset.

4.3.1 The Multi-choice Question Answering in Examinations Shared Task

The Multi-choice Question Answering in Examinations Shared Task 2017 (IJCNLP Shared Task Nº 5) (**MCQA**) took place at The 8th International Joint Conference on Natural Language Processing (IJCNLP 2017).[11] A detailed description of the shared task is provided by Shangmin et al. (2017). This is a typical question answering task that aims to test how accurately answers to exam questions can be selected. The dataset provides examination questions in two languages: English and Chinese. I focus on the English subset which contains 5,367 questions from five domains: Biology, Chemistry, Earth Science, Life Science and Physical Science. Table 4.7 presents the division of the dataset into training, development, and test sets. Examples of question and answer candidates are shown in Table 4.8. The questions come from real exams. Every question has four answer candidates which may be a word, a value, a phrase or a sentence. The authors of the dataset did not provide any supporting documents and allow participants to choose their own additional resources.[12] Any additional sources, such as knowledge bases, Wikipedia, textbooks, or articles, can be used to help answer the questions.

Domain	**Train**	**Val**	**Test**	**Total**
Biology	281	70	210	561
Chemistry	775	193	581	1549
Physical	299	74	224	597
Earth Science	830	207	622	1659
Life Science	501	125	375	1001
Total	2686	669	2012	5367

Table 4.7: The number of questions for each domain and the training, development and test devision of the English subset of the MCQA dataset.

[11] `http://www.nlpr.ia.ac.cn/cip/ijcnlp/Multi-choice_Question_Answering_in_Exams.html` – last verified May 2021

[12] This is why the MCQA dataset is not presented in Chapter 2.

Chemistry	Life-science
Factoid Question:	**Cloze Style Statement Question:**
Which two elements, when combined, are NOT alloys?	Regular black coffee without any sweetener would taste _____ to many people.
– iron and carbon	– sweet
– copper and lead	– salty
– carbon and sulfur	– sour
– **antimony and iron**	– **bitter**

Table 4.8: An example of question and cloze question, and answer candidates from the MCQA dataset. Bold marks the correct answer.

4.3.2 Experiments

The main difference between this system and the system applied to the MovieQA dataset is the data selection module: in the former work, I select sentences from a movie plot based on similarities with the question and provided answers; here the relevant sentences are extracted from Wikipedia based on key words. The full approach is illustrated in Figure 4.4.

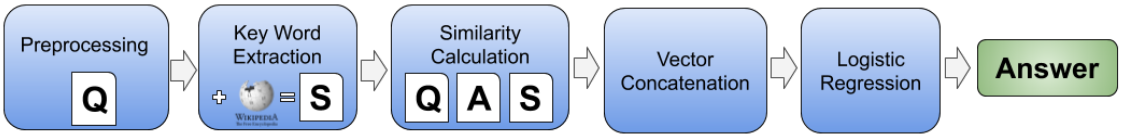


Figure 4.4: The approach pipeline for MCQA dataset.

4.3.2.1 Prepossessing and Data Selection

The first step is to clean the question and answer text. I remove serial numbers and letters from the question and answer using regular expressions. See Examples (4.17) - (4.19).

(4.17) *"3. What ..." → "What ... "*

(4.18) *"8) Who ..." → "Who ... "*

(4.19) *"a) Paper: Paper degrades ... " → "Paper: Paper degrades ..."*

Although the authors of the dataset do not provide additional text, relevant sentences can be extracted from Wikipedia via Lucene[13] as was also done in the baseline. Instead

[13] `https://lucene.apache.org/core/` – last verified December 2021

of searching for the entire question with an answer candidate I looked for key words. A list of keywords are extracted from the question statement using the Natural Language Toolkit (nltk)[14] implementation of the Rapid Automatic Keyword Extraction (RAKE) algorithm proposed by Rose et al. (2010). RAKE is an unsupervised, domain-independent algorithm, which determine the words and phrases based on the frequency of word appearance and its co-occurrence with other words in the text.

RAKE as an input takes a number of predefined words and punctuation which are used as delimiters for other words and phrases in a document. It can be stop words and other words with minimum lexical meaning. First RAKE splits the text using those delimiters and builds the keyword candidate from individual words and a sequence of contiguous words. Then keyword candidates are scored based on co-occurrence with all other words in the text. As some multi-word phrases may contain stop words inside (e.g. set *of* numbers), the algorithm joins the candidates together. Finally, the top rated words (phrases) are returned as key words (phrases).

The extracted key words are used to retrieve a list of sentences from Wikipedia. For each question, the top 50 sentences ranked by (unweighted) keywords related to the item are selected. I tried both searching for key words of the question only and searching for key words of the question and answer candidates. The second option produced better results. When obtaining information in this way, there is no guarantee the extracted sentences will contain the answer and, in some cases, they do not. Table 4.9 gives examples of "good" and "bad" passages for two questions.

The systems for the answer selection are also built based on the string similarities presented in Section 3.2 and logistic regression. Five types of string similarity are used: `w2v_cos`, `cos_tfidf`, `bow`, `wSlide`, `charN_gramm` (N = 2,3,4,5). Those five similarities give seventeen features: eight of them are calculated between the sentences S and the answer candidate a, another eight are calculated between the sentences S and the concatenation of the question and the answer candidate $q + a$. Another feature is `Word2Vec`

[14] `https://github.com/csurfer/rake-nltk` – last verified December 2021

Passage Contains the Answer:	**Passage without Answer:**
Life-science	**Physical-science**
Question: Mitosis followed by division results in	An example of a general scientific publication is
– two genetically different cells. – **two genetically identical cells.** – two cells with half as much DNA as the parent cell. – four cells with half as much DNA as the parent cell.	– "?Science.?" – "?Journal of Applied Physics.?" – "?American Journal of Physics.?" – **"?Journal of the American Chemical Society.?"**
Combined Key Words: mitosis followed, division results, two genetically different cells, two genetically identical cells, two cells, parent cell, much dna, half, parent cell, four cells	general scientific publication, example, science, applied physics, journal, american journal, physics american chemical society
Passage: Organelles may also be split between two cells during the process of cellular division ... **In mitosis, one cell divides to produce two genetically identical cells.** ...	(1956) in applied physics from Harvard University. science. journal. in applied physics and electronics in 1969 from University of Rajshahi. candidate in applied physics. example

Table 4.9: Examples of sentences extracted from Wikipedia concatenated together, which contain and do not contain the answer. Bold marks the correct answer and the part of text relevant to question. Original formatting is kept.

sentence-level similarity. It is calculated as shown in (4.20):

$$w2v\_sl(q, a, S) = max_{s \in S}(w2v\_cos(q, s) + w2v\_cos(a, s)) \tag{4.20}$$

where S is a set of relevant sentences, $w2v\_cos$ is word2vec cosine similarity.

Equation (4.20) is a special case of Equation (4.4). Table 4.10 shows all the 17 similarities that were used. If the question q includes gaps, instead of concatenating, the candidate answer a will be used to fill the gaps in the question, as described in Section 4.1.2. See Example (4.21).

(4.21) **Question:** *"_____ obtain energy by using the chemical energy stored in inorganic compounds"*

Answer candidates –> concatenation strings:

1. *Photoautotrophs –> Photoautotrophs obtain energy by ...*

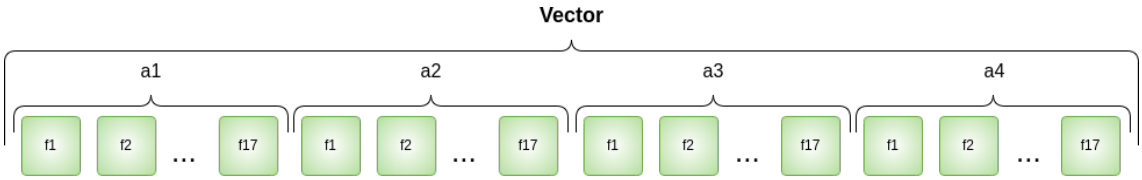


Figure 4.5: Feature vector concatenation.

2. *Chemoautotrophs –> Chemoautotrophs obtain energy by ...*

3. *Heterotrophs –> Heterotrophs obtain energy by ...*

4. *None of the above –> None of the above obtain energy by* $< ... >$

List of used features for MCQA dataset	
$f_1 = w2v\_sl(q, a, S)$	
$f_2 = w2v\_cos(a, S)$	$f_{10} = w2v\_cos(q + a, S)$
$f_3 = cos\_tfidf(a, S)$	$f_{11} = cos\_tfidf(q + a, S)$
$f_4 = bow\_overlap(a, S)$	$f_{12} = bow\_overlap(q + a, S)$
$f_5 = windowSlide(a, S)$	$f_{13} = windowSlide(q + a, S)$
$f_{6-9} = charNgram(a, S)$ *N=2,3,4,5*	$f_{14-17} = charNgram(q + a, S)$

Table 4.10: List of all used features for MCQA dataset.

For a question q the similarity features of its answer candidates a_1, a_2, a_3, a_4 are concatenated into one single vector as shown in Figure 4.5. Following this method, for each question, there is one feature vector which contains information for all answer candidates inside. The final step of the system is the logistic regression over the vector.

4.3.2.2 Baseline

To make sense of results comparison, first, I describe a baseline provided by Shangmin et al. (2017). It involves concatenating the query with the answer candidates and extracting relevant documents from Wikipedia[15] using the open-source information retrieval library Lucene. Then the authors score the documents based on similarity with the query, choosing three or fewer of the highest scored documents to calculate the score for each candidate answer.

[15] `https://dumps.wikimedia.org/` – last verified May 2020

4.3.2.3 Results

Considering the different domains there are two ways to train the model:

- **Combined training:** Data from all domains is combined together to train one single model.
- **Domain training:** Use separate models for each domain. The hyper-parameters of the logistic regressions are the same for every domain.

The results obtained by these systems and the baseline are presented in Tables 4.11 and 4.12. The correct answers are not equally spread across answer candidates. The third answer is correct in 36% of all the data. So the majority class prediction would be 42.75% for the development set and 30.81% for the test set. Using the combined training approach, 43.6% on the development set was obtained. This result outperforms the baseline. When training on an individual domain it is observed that the average score is better than combined training. The best performance is obtained in the Chemical subset – 64.77% accuracy on the development set and 64.03% on the test set. However, for Earth Science and Life Science, the accuracy is 30.43% and 32.80% on the development set and 32.32% and 32.53% on the test set respectively. These numbers are below the baseline system result for the development set.

System	Dev set	Test set
Baseline	29.45	–
Majority class	42.75	30.81
Combined training	43.60	–
Domain training	**48.70**	**45.60**

Table 4.11: Results of all English subset of the baseline system on the development set, majority class, and my system for combined and domain training on the development and test sets.

4.3.3 Discussion

In this section I place my results in context with other work on exam QA.

	Valid		Test	
	Baseline	**Domain training**	**Baseline**	**Domain training**
Biology	30.00	**47.14**	-	45.23
Chemistry	21.24	**64.77**	-	64.03
Physics	25.68	**41.89**	-	39.73
Earth science	**31.88**	30.43	-	32.32
Life science	**40.00**	32.80	-	32.53

Table 4.12: Results of the baseline system and domain training on the development set and test set for each domain.

Wang et al. (2014) describe CMU's UIMA-based[16] modular automatic QA system to automatically answer multiple-choice questions for the entrance exam in world history in English and Japanese.[17] The approach relies on two different test collections: the original test collection provided by NTCIR[18] organizers and the collection created by the authors.

Li et al. (2013) describe the system that was used in the Entrance Exams task of Question Answering for Machine Reading Evaluation on CLEF-2013 (QA4MRE CLEF) (Peñas et al., 2013). It consists of three components: Character Resolver, Sentence Extractor and Recognizing Textual Entailment. Documents are processed by the Character Resolver in order to detect all story characters, tag each story with a character as an ID, and resolve personal pronouns. The Sentence Extractor then extracts related sentences for each question and creates a Hypothesis (H) and Text (T). Finally, it inputs this T/H pair into the Recognizing Textual Entailment system to select an answer, which uses logical representations based on WordNet (Fellbaum and Miller, 1998).

Table 4.13 presents the results of other systems who participated in the Shared Task 2017.[19]

I obtain the best performance of 45.6% accuracy. The second best result of 42.2% is obtained by a CNN-LSTM Model with Attention proposed by Wang et al. (2017a). The authors first learn a joint representation of question-answer candidate pairs with a CNN. Then they use it as an input into a LSTM with attention to learn a binary labeling

[16] `https://uima.apache.org/` – last verified September 2021

[17] The data is based on `https://www.regentsprep.org/` – last verified May 2020

[18] `http://research.nii.ac.jp/ntcir/index-en.html` – last verified September 2017

[19] `http://159.226.21.226/QAtask/score_list/` – Shared Task Leader Board. The described system is "Cone" – last verified January 2018

System	Dev set	Test set	Extra Data	Similarity Usage
Baseline (Shangmin et al., 2017)	29.45	–	✓	✓
MappSent (Hazem et al., 2017a)	34.1	30.3	✗	✓
YNU-HPCC (Yuan et al., 2017)	34.5	35.5	✗	✗
JU NITM (Sarkar et al., 2017)	40.7	40.6	✗	✓
YNUDLG (Wang et al., 2017a)	39.6	42.2	✓	✗
My work (Dzendzik et al., 2017a)	**48.7**	**45.6**	✓	✓

Table 4.13: Results of all teams on English subseton the development (**dev**) and **test** sets.

for each question-answer option. As an additional source of data, the authors used 5 scientific question-answering datasets from the Allen Institute for Artificial Intelligence:[20] AI2 Science Questions v2,[21] Textbook Question Answering (Kembhavi et al., 2017), SciQ dataset (Welbl et al., 2017), Aristo MINI Corpus (Clark et al., 2018), and StudyStack Flashcards.[22] Sarkar et al. (2017) based their method on a decision tree classifier and using word embedding features (word2vec trained on GoolgeNews[23]). This approach achieved 40.7% and 40.6% accuracy on the development and test set. Similar to the work of Wang et al. (2017a), Yuan et al. (2017) also used an attention-based LSTM model. The authors also compared the use of two embeddings for this task: word2vec trained on GoogleNews and the GloVe trained on Twitter. The same model was applied for both the English and Chinese version of the dataset. The authors claim they extended the training set with examination question examples from the Internet without providing further details. Finally, Hazem et al. (2017a) apply a textual similarity approach originally proposed for question-to-question similarity (Hazem et al., 2017b).

Three out of six systems, including the baseline and this work, used additional data sources for answering the questions. The two non-baseline systems show better results than the others. That means that questions can be answered better with the additional data rather than without it. Also, four systems, including the baseline and this work, use different versions of string similarity. It would appear the combination of string similarities

[20] `http://allenai.org/data.ht` – last verified May 2020
[21] `http://data.allenai.org/ai2-science-questions/` – last verified May 2020
[22] `https://www.studystack.com/`– last verified May 2020
[23] `https://code.google.com/archive/p/word2vec/` – last verified May 2020

and additional data sources is beneficial for this type of question answering. To the best of our knowledge, there are no further studies using the English version of the dataset.

MCQA also contains a group of questions which my method is not designed to answer. Those questions have an option *"all of the above"* and *"none of the above"*, and it is always the last option. At the core of my method is string similarity but those options have only a logical connection established not through vocabulary or semantics. The proposed similarity-based approach focuses on selecting the particular answer and does not contain any component which might perform aggregation over the options.

4.4 Summary

In this chapter I introduced an approach for multiple choice reading comprehension question answering task based on a combination of string similarities and logistic regression. I tested my method on two datasets available at the time and obtained state-of-the-art results.

As I can see from experiments with the MovieQA dataset, the method gains a lot from using `TF-IDF` similarity. This could mean that if the questions and answer candidates were formulated with a richer more diverse vocabulary and without repetition of words from the plot, my method might not work so well. It could be applied in similar real life scenarios, such as in a narrow domain with limited, well defined terms, and a limited vocabulary.

The advantage of the proposed method is computational complexity. All experiments can be run within 24 hours on any Central Processing Unit (CPU) without usage of a Graphics Processing Unit (GPU). The heaviest part of the experiment was calculating the `SkipThoughts` representations. This component of the current system setup can be removed without hurting performance.

This work was carried out in 2016-2017 and would not compete with the state-of-the-art systems based on BERT or more complex neural networks. However, it could be used for building a fast prototype or establishing an easy-to-interpret baseline. Another usage

of the described approach could be in the case of limited computational resources.

In the case of MCQA, the key component was the additional data. The method I used to extract relevant sentences from Wikipedia is simple. However, that is something the other shared task participants did not do, as some of them used data as is, while other looked for extra question-answer pairs rather than evidence of the correct answer.

To sum up, my main findings are:

1. Logistic Regression based on string similarities produced results comparable to the results obtained with more complex neural networks as CNN and LSTM on the datasets where the question and answer candidates are formulated with the same vocabulary as related passage;

2. Additional sentence-level features with a combination of selected sentences improve the results;

3. On the MovieQA dataset the main source of error is rephrasing and the fact that the information required for the answers is spread over multiple sentences;

4. On the Shared Task data, the main challenge and source of error is a lack of supported texts.

The result of this work has been published at The Joint Video and Language Understanding Workshop: MovieQA and The Large Scale Movie Description Challenge (LSMDC), at ICCV 2017,[24] (Dzendzik et al., 2017b), and work on MCQA was published in Shared Task 5: Multi-choice Question Answering in Examinations at The 8th International Joint Conference on Natural Language Processing (IJCNLP 2017) (Dzendzik et al., 2017a).[25]

[24] https://drive.google.com/file/d/0B9nOObAFqKC9WEE5cTJYSzNZODQ/view?resourcekey=0-q_HG8dCdOBi-8abry4FVyA – last verified November 2021

[25] https://aclanthology.org/I17-4010/ – last verified November 2021

Chapter 5

Answering Boolean Questions

> "Exactly!" said Deep Thought. "So once you do know what the question actually is, you'll know what the answer means."
>
> *Douglas Adams*
> *The Hitchhiker's Guide to the Galaxy*

Boolean questions imply the answer should be binary: *"Yes"* or *"No"*. A brief definition and discussion of boolean question and datasets was provided in Chapter 2. At first glance answering boolean questions could be considered to be one of the easiest QA tasks as there are only two possible options for the answer. But with detailed consideration there is much understanding involved in answering boolean questions. The work of Clark et al. (2019) explores the difficulty of Yes/No questions and introduces the BoolQ dataset.

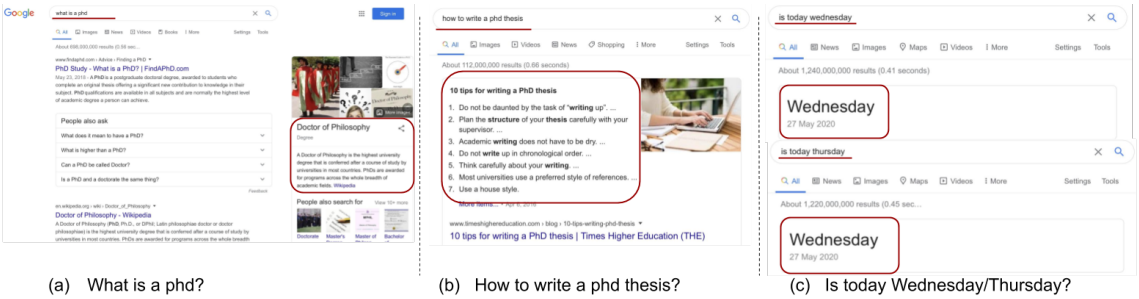
In this chapter I focus on user generated questions and, in particular, examine the BoolQ dataset. Part of the work described in this chapter was carried out during an internship at Google Research Switzerland between September and December 2019 in collaboration with Massimo Nicosia.

This chapter is structured as follows: first I discuss the motivation for answering

boolean questions in Section 5.1. Then, in Section 5.2, I look in detail at the BoolQ dataset. After that, in Section 5.3, I discuss baseline results, re-productivity issues, and the relation between passage length and accuracy, followed by an error analysis. I summarise my main findings in Section 5.4

5.1 Motivation

Search engines have become much more than a tool for searching for general information, they are also capable of providing structured data and directly answering some questions. For example, for some factoid questions there is an info box with the definition, or for some instructional queries there is an extracted list of steps or advice. For the question *"What is a PhD?"* the Google search engine returns a definition of a Ph.D. in the right side of the page and for the query *"How to write a PhD?"* is returns a numbered list of tips (see Figure 5.1). That works for some simple boolean questions as well. For example for both queries *"Is today Wednesday?"* and *"Is today Thursday?"* the search engine provides the same info box telling the day of the week and the date.[1]



(a) What is a phd? (b) How to write a phd thesis? (c) Is today Wednesday/Thursday?

Figure 5.1: Example of structured answer from search engine Google.ie. The red box highlights the structured answer.

However, examining different queries and the way they are formulated I observe different results. Figure 5.2 illustrates how a search result changes according to the question formulation. Asking the factoid question *"Where do pandas live?"* I get the answer *China* (Figure 5.2(a)). The questions *"Do pandas live in China?"* and *"Do pandas only live in China?"* do not give a direct answer but provide enough accurate and suitable informa-

[1]Both queries were entered on Wednesday, the 27th of May 2020.

tion to conclude the answer is *yes* for both of them (see Figure 5.2 (b) and (c)). But the answer to the question *"Do pandas live in Russia?"* (Figure 5.2 (e)) again does not provide the direct answer and refers to Moscow Zoo which implies Russia hosts pandas in the Moscow Zoo but does not provide information on whether pandas live in the territory of Russia. Then, if I try other boolean questions about pandas and different countries I obtain three different results: Figure 5.2 (d): *"Does Japan have pandas?"* – the answer contains the explicit names of pandas who live in Japan and the answer *"Yes"* can be easily concluded. Figure 5.2 (f): *"Does Russia have pandas?"* – the answer lists a number of countries which have pandas in the zoo and Russia is one of them, so without a particular specification the answer *"Yes"* still can be deduced. Figure 5.2 (g): *"Does Ireland have pandas?"* – the search result refers to an event which happened in 1986 and contains enough information to conclude the answer *"No"*. None of the provided boolean questions, regardless of the form or arguments (countries in this case), was directly answered. This suggests that there is room for improvement and that the task of boolean QA is worth future investigation.

As was shown in Chapter 2, more then a third of the considered datasets have boolean questions in some form (see Figure 2.4 in Chapter 2). Another wide application of answering boolean questions is product/service related questions asked by users on all kind of forums and recommendation websites. For example, the AmazonQA dataset contains a significant portion (15%) of yes/no questions. I will talk about answering user boolean questions about products in detail in Chapter 7.

5.2 BoolQ Dataset

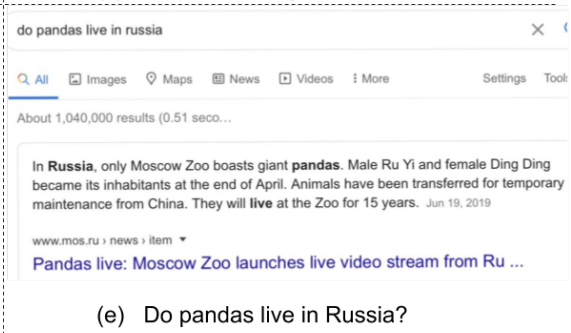
In this section I look in detail at the BoolQ dataset and the baseline system provided by Clark et al. (2019). The dataset contains 15942 questions based on real user Google queries and a passage from Wikipedia assigned to each question by crowdsourcers. About 3000 questions and passages come from the NaturalQuestions dataset (Kwiatkowski et al., 2019). The dataset is unbalanced: around 62 % of questions have the answer "Yes".

(a) Where do pandas live?

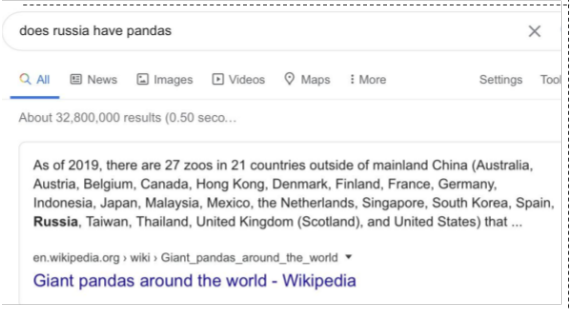
(b) Do pandas live in China?

(c) Do pandas only live in China?

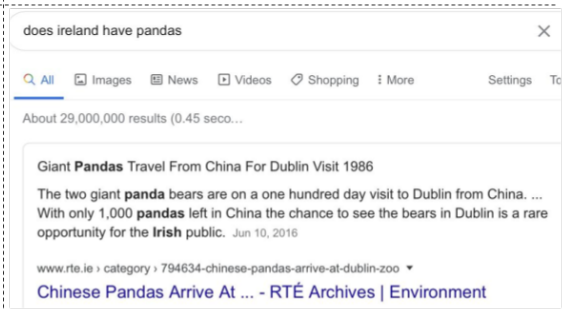
(d) Does Japan have pandas?



(e) Do pandas live in Russia?



(f) Does Russia have pandas?



(g) Does Ireland have pandas?

Figure 5.2: Example of different form of answer from search engine Google.ie

BoolQ has been included in the SuperGLUE benchmark[2] (Wang et al., 2019). Dataset statistics are provided in Table 5.1

[2] `https://super.gluebenchmark.com/` – last verified May 202‘

Size	Division			Length in Tokens					
				Question			Passage		
	Train	Dev	Test	Min	Max	Avg	Min	Max	Avg
15942	9.4k	3.2k	3.2k	3	21	8.9	6	813	108

Table 5.1: BoolQ statistics.

5.2.1 A Closer Look at the Data

The questions are quite varied. Some of the questions are straightforward and can be directly answered from the text, like Examples (5.1) and (5.2).[3] In the first case, the answer "*Yes*" is explicitly explained in the passage and can be inferred (as *cream* is a type of *dairy*). Here and in following examples I mark the part of passage relevant to the question in bold. In the second example, based on the last sentence, the answer "No" can be concluded.

(5.1) **Question:** *does penne alla vodka have dairy in it*

Passage: *Vodka sauce is an Italian-American cuisine sauce made from a smooth tomato sauce, vodka, typical Italian herbs and* ***heavy cream****, which gives the sauce its distinctive orange coloration. ...*

Answer: Yes

(5.2) **Question:** *does maple syrup come straight from the tree*

Passage: *Maple syrup is a syrup usually made from the xylem sap of sugar maple, red maple, or black maple trees, ... and collecting the exuded sap,* ***which is processed by heating to evaporate much of the water, leaving the concentrated syrup****.*

Answer: No

Other questions require some additional knowledge and/or summarising the passage information. For example to answer question "*Does Damon and Elena get together in season 3?"* in (5.3) the whole long passage has to be analysed combining together multiple indirect statements (see bold text in the passage) and world knowledge. Another example is (5.4), where the passage does not mention *Smeagol* and uses the other name

[3]Here and in all following examples I keep the original BoolQ format: the questions are lower cased and there is no question mark at the end.

Gollum, so additional knowledge that *Smeagol* is actually *Gollum* is required. Also, the passages does not directly say the character died, instead it says "*fell into the fires of the volcano where both he and the Ring were destroyed*" which implies death and that the answer is "Yes".

(5.3) **Question:** *does damon and elena get together in season 3*

Passage: *In the third season, ... A still loyal* ***Elena, however, refuses to admit her feelings for Damon.*** *In 'Dangerous Liaisons', Elena, frustrated with her feelings for him, tells Damon that* ***his love for her may be a problem, and that this could be causing all their troubles.*** *... in a moment of heated passion,* ***Elena – for the first time in the three seasons – kisses Damon*** *of her own accord. This kiss finally causes Elena to admit that* ***she loves both brothers*** *and realize that she must ultimately make her choice ... She chooses the latter when* ***she calls Damon to tell him her decision. Damon****, who is trying to stop Alaric,* ***accepts what she says and she tells him that maybe if she had met Damon before she had met Stefan, her choice may have been different.*** *...*

Answer: No

(5.4) **Question:** *does smeagol die in lord of the rings*

Passage: *The Ring, which Gollum referred to as "my precious" or "precious", extended his life far beyond natural limits. Centuries of the Ring's influence twisted Gollum's body and mind, and, by the time of the novels, he "loved and hated (the Ring), just as he loved and hated himself." Throughout the story, Gollum was torn between his lust for the Ring and his desire to be free of it. Bilbo Baggins found the Ring and took it for his own, and Gollum afterwards pursued it for the rest of his life. Gollum finally seized the Ring from Frodo Baggins at the Cracks of Doom in Orodruin in Mordor;* ***but he fell into the fires of the volcano, where both he and the Ring were destroyed.***

Answer: Yes

Some questions are formulated in such a way that their answer might change over

time, e.g, Example (5.5) which is asking about a movie released *this year*. As the dataset was released in 2019 the data could be collected in 2018 so then the answer is *yes* but if this question would be asked in 2015 or last year (2020) the answer should be *no*.

(5.5) **Question:** *is there a star wars movie this year*

Passage: *The first film was followed by two successful sequels, The Empire Strikes Back (1980) and Return of the Jedi (1983); ... A prequel trilogy was released between 1999 and 2005, albeit to mixed reactions from critics and fans. A sequel trilogy concluding the main story of the nine-episode saga began in 2015 with The Force Awakens. ... Together with the theatrical spin-off films The Clone Wars (2008), Rogue One (2016) and Solo: A Star Wars Story (2018), Star Wars is the second highest-grossing film series ever.*

Answer: Yes

Another example is (5.6) where a passage provides information about United States citizens border crossing requirements, and the specific details about citizenship is obtained not from the passage itself but from the title.[4] However, the question does not specify what kind of citizenship the person, who is asking the question, holds. In contrast, the question from Example (5.7) can be answered unconditionally (without additional assumptions) as a holder of the Schengen visa (information from the question) can enter Montenegro for 30 days (information from the passage). So, in such cases like Examples (5.5) and (5.6), the question and the passage do not contain enough information to answer the questions unconditionally. As they are ambiguous.

(5.6) **Question:** *can i get into canada with a military id*

Passage: *(Title: American entry into Canada by land) Canadian law requires that all persons entering Canada must carry proof of both citizenship and identity. A valid U.S. passport or passport card is preferred, although a birth certificate, naturalization certificate, citizenship certificate, or another document proving U.S.*

[4]The title of the article is a part of dataset but was not used in the baseline system. So it can be considered as meta information.

nationality, together with a government-issued photo ID (such as a driver's license) are acceptable to establish identity and nationality.

Answer: Yes

(5.7) **Question:** *can i go to montenegro with a schengen visa*

Passage: *Nationals of any country may visit Montenegro without a visa for up to 30 days if they hold a passport with visas issued by Ireland, a Schengen Area member state, ...*

Answer: Yes

Some passages appear to be irrelevant or do not contain enough information to obtain the answer. The passages in (5.8) and (5.9) are related to the questions but specific information is missing and the answer "Yes" cannot be confirmed by the passages.

(5.8) **Question:** *is daisy the director of shield in the comics*

Passage: *Daisy Johnson, also known as Quake, is a fictional superhero appearing in American comic books published by Marvel Comics. Created by writer Brian Michael Bendis and artist Gabriele Dell'Otto, the character first appeared in Secret War #2 (July 2004). The daughter of the supervillain Mister Hyde, she is a secret agent of the intelligence organization S.H.I.E.L.D. with the power to generate earthquakes.*

Answer: Yes

(5.9) **Question:** *is chicken cordon bleu made with blue cheese*

Passage: *A cordon bleu or schnitzel cordon bleu is a dish of meat wrapped around cheese (or with cheese filling), then breaded and pan-fried or deep-fried. Veal or pork cordon bleu is made of veal or pork pounded thin and wrapped around a slice of ham and a slice of cheese, breaded, and then pan fried or baked. For chicken cordon bleu chicken breast is used instead of veal. Ham cordon bleu is ham stuffed with mushrooms and cheese.*

Answer: Yes

I also spotted some errors in the data, for example (5.10) - (5.12). The first example is asking if shower gel can be used instead of shampoo in a negative form (*"is it bad to ..."*) and the passage says that they are perfectly substitutable so the answer should be *No (it is not bad)*. In the second example (5.11), the passage explicitly says India does not have a national language so the answer should be "*No*". In the third Example (5.12) the question does not explicitly refer to the 2018 FIFA World Cup but Russia hosted the World Cup only once in 2018. There is nothing in the passage that should make the reader believe there were any games outside of Russia, so the answer should be "*Yes*".

(5.10) **Question:** *is it bad to wash your hair with shower gel*[5]

Passage: *... This means that* ***shower gels can also double as an effective and perfectly acceptable substitute to shampoo****, even if they are not labelled as a hair and body wash.*

Answer: Yes, Should be **No**

(5.11) **Question:** *is hindi is my national language of india*

Passage: *The Constitution of India designates the official language of the Government of India as Hindi written in the Devanagari script, as well as English.* ***There is no national language as declared by the Constitution of India.*** *Hindi is used for official purposes such as parliamentary proceedings, ...*

Answer: Yes, Should be **No**

(5.12) **Question:** *are all world cup matches played in russia*

Passage: *The 2018 FIFA World Cup was the 21st FIFA World Cup, an international football tournament contested by the men's national teams of the member associations of FIFA once every four years. It* ***took place in Russia*** *from 14 June to 15 July 2018. It was the first World Cup to be held in Eastern Europe, and the 11th time that it had been held in Europe.*

Answer: No, Should be **Yes**

[5]We keep the original spelling from the dataset

5.2.2 Types of Questions and Reasoning

To provide a better understanding of the data, Clark et al. (2019) analysed 110 randomly selected samples from the BoolQ dataset and divided it by question topic (*Nature and Science, Law and Government, Sports, Entertainment and Media, Fictional Events, History*, and *Other*), question type (*Definitional, Existence, Event Occurrence, Other General Fact*, and *Other Entity Fact*), and type of reasoning (*Paraphrasing, By Example, Factual Reasoning, Implicit, Missing Mention*, and *Other Inference*).

I add two more question topic categories:

- **Cooking and Food** – questions including ingredients of dishes and properties of food, such as Examples (5.1), (5.9) and (5.13)-(5.15):

  (5.13) *is a tablespoon bigger than a dessert spoon*

  (5.14) *is there raw egg in egg drop soup*

  (5.15) *does a fried egg have a runny yolk*

- **Geographic Location** is a topic about everything that is related to countries geography, including time zone, e.g. (5.16)-(5.17) and existence of certain objects in certain places on Earth e.g. (5.18).

  (5.16) *can you get to south america by car*

  (5.17) *is france the same timezone as the uk*

  (5.18) *is there a waterfall in marble falls tx*

5.2.2.1 Question Type

Clark et al. (2019) divide questions into five types, and I supplement this list with the type *Possibility*:

- **Definitional** is a type of question asking about alternative names for objects or concepts, or whether an object belongs to a specific category. An example of such a question is shown in (5.19)-(5.20):

(5.19) *is dragon fruit and pitaya the same thing*

(5.20) *is a half barrel the same as a keg*

- **Existence** is a type of question which asks about the existence of a particular object or a concept with particular properties, e.g. (5.21) and (5.22):

  (5.21) *is there an age limit to compete in the olympics*

  (5.22) *is there a movie after i am number 4*

- **Event Occurrence** is a type of question asking about whether a particular event took place, e.g. (5.23) and (5.24):

  (5.23) *do donna and eric end up getting married*

  (5.24) *does mr darcy die in pride and prejudice and zombies*

- **Other General Fact**, e.g. (5.25):

  (5.25) *do you have to break in new car engines*

- **Other Entity Fact** (e.g. (5.26)):

  (5.26) *is the bronx zoo the largest zoo in the world*

- **Possibility** is a type of question asking about a practical or hypothetical possibility of some event or action. Usually such questions start with the word *"Can"*, e.g. (5.27)-(5.29).

  (5.27) *can you drive a us car into canada*

  Answer: *Yes*

  Passage: *Persons driving into Canada must have their vehicle's registration document and proof of insurance.*

  (5.28) *can i perform a copyrighted song in public*

  Answer: *No*

  Passage: *Permission to publicly perform a song must be obtained from the copyright holder or a collective rights organization.*

Such questions can be tricky to answer. Example (5.29) is about the possibility for the African team to win a football competition. The answer is *yes* as every time some teams from the African continent participate in the FIFA World Cup. Although it has never happened,[6] theoretically it is possible and according to the passage nothing prevents this event from happening.

(5.29) *can an african nation win the world cup*

Answer: *Yes*

Passage: *Since the second World Cup in 1934, qualifying tournaments have been held to thin the field for the final tournament. They are held within the six FIFA continental zones (Africa, Asia, North and Central America and Caribbean, South America, Oceania, and Europe), overseen by their respective confederations. For each tournament, FIFA decides the number of places awarded to each of the continental zones beforehand, generally based on the relative strength of the confederations' teams.*

5.2.2.2 Reasoning Type

Clark et al. (2019) define six categories based on the type of reasoning required to answer the questions:

- **Paraphrasing** – the answer to the question is explicitly in the passage but could be formulated differently.
- **By Example** – there is an example (or counter-example) of an event or situation in the passage which leads to the answer.
- **Factual Reasoning** – some external world knowledge is needed in addition to the passage to answer the question.
- **Implicit** – the passage is formulated in such a way that the particular answer can be concluded because the opposite answer would not make sense.

[6] According to the Wikipedia Page `https://en.wikipedia.org/wiki/FIFA_World_Cup` – last verified May 2021

- **Missing Mention** – the question is formulated in such a way to imply that certain information should be present. The lack of this information leads to the opposite answer, e.g. Example (5.30) where teams came back from 3-0 in MLB and NHL but not in NBA.

  (5.30) **Question:** *has anyone ever came back from 3-0 in nba*

  Passage: *The following is the list of teams to overcome 3–1 series deficits by winning three straight games to win a best-of-seven playoff series. In the history of major North American pro sports, teams that were down 3–1 in the series came back and won the series 52 times, more than half of them were accomplished by National Hockey League (NHL) teams. Teams overcame 3–1 deficit in the final championship round eight times, six were accomplished by Major League Baseball (MLB) teams in the World Series. Teams overcoming 3–0 deficit by winning four straight games were accomplished five times, four times in the NHL and once in MLB.*

  Answer: No (otherwise it would be mentioned)

- **Inference Other** – there is information in the passage which does not provide the answer directly but the answer can be concluded with some reasoning e.g. excluding other entity properties. Here I use the same example which was provided by (Clark et al., 2019, p.2928), see Example (5.31):

  (5.31) **Question:** *is the sea snake the most venomous snake*

  Passage: *... the venom of the inland taipan, drop by drop, is the most toxic among all snakes*

  Answer: No. [If inland taipan is the most venomous snake, the sea snake must not be.]

5.3 BERT-based Baseline System

Clark et al. (2019) established a strong baseline using the $BERT_{large}$ model (Devlin et al.,

2019), which outperforms the recurrent ELMO model (Peters et al., 2018). Also Clark et al. (2019) showed that transfer learning from an entailment task improves the results by 4.11%. The model was pre-trained on The Multi-Genre Natural Language Inference Task (MultiNLI or MNLI) (Williams et al., 2018) and then fine tuned on BoolQ.

5.3.1 Reproducibility

I attempted to reproduce the results of the baseline $BERT_{large} + MNLI$ model released by Clark et al. (2019).[7] Its accuracy is between 80% and 82% (Fig. 5.3 (a) ●) with an average 81.41% accuracy over 10 runs (vs. 82.2% reported in Clark et al. (2019)). The error analysis shows a significant number of the correctly answered questions varies from run to run. The average number of errors is 607, while only 366 errors are common, which means that 40% of errors are fluctuating.

I define the ratio of the number of correctly answered questions across n runs to the total number of questions as *stable accuracy*. Formally, if Q is the set of all questions and $Q^i_{correct}$ is the set of correctly answered questions at the i^{th} run, the *stable accuracy* after n runs is defined as (5.32):

$$StableAccuracy_n = \frac{|\cap_{i=0}^{n} Q^i_{correct}|}{|Q|} \tag{5.32}$$

The stable accuracy over 10 runs drops to 71% (see Fig 5.3 (a) ★). Ensembling with a majority voting for up to 10 runs (Fig. 5.3 (a), ▲) does not outperform the baseline: the values are within the range of 78.09% and 81.77%.[8]

I repeated the experiment using the robustly optimized BERT model $RoBERTa_{large}$ (Liu et al., 2019b) implemented by Wolf et al. (2020) with PyTorch,[9] and fined tuned on MNLI task. This model has a better average accuracy (83.7)% but it is also more

[7] `https://github.com/google-research/language/tree/master/language/boolq` – last verified May 2020

[8] Note that the ensemble performs slightly better with an odd numbers of runs as only the samples with strictly more votes for the correct answer are considered to be answered correctly. This is a very strict evaluation. Alternatively, in the case of a tie, the majority answer (*Yes*) or randomly chosen answer can be selected, but I aim to provide the evaluation with the maximum certainty.

[9] `https://pytorch.org/` – last verified May 2021

unstable: the stable accuracy drops to 64.0% (see Fig. 5.3 (b)). As with the $BERT$ model, ensembling over 10 runs does not give a performance boost.

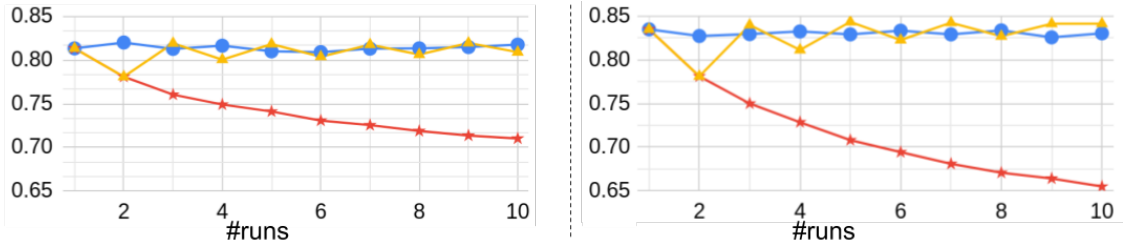


Figure 5.3: (a) BERT and (b) RoBERTa based implementation of the baseline. Accuracy (blue), stable accuracy (red), and majority voting accuracy (yellow) over 10 runs.

This observed behavior means that the system performs well on each run but every time it performs well on a different set of questions. This might be related to the notion of "forgettable" examples described by Toneva et al. (2019).[10]

The difference is that they discovered the ability of models to forget the learned examples during the training phase, while we examine stable and unstable examples when the training is finished.

This unstable behavior of $BERT$ and $RoBERTa$ models was discovered unintentionally. Simultaneously to this work, McCoy et al. (2020) show that when BERT is fine-tuned on the MNLI training set and tested on the MNLI development set, its behavior is relatively unstable. However, the same models can significantly vary in out-of-domain linguistic generalisation when applied to the HANS dataset (McCoy et al., 2019). In my experiments I observe that both $BERT$ and $RoBERTa$ models, fine-tuned on BoolQ, do not generalize to the BoolQ (in-domain) test/development set. There could be a few reasons for that: (1) the passages in BoolQ are relatively long; (2) the BoolQ dataset is relatively small (the size of MNLI development set is almost the same as the size of BoolQ training set).

[10]Toneva et al. (2019) explored the training dynamics of neural network on picture classification tasks. They define an example as "forgettable" if it has been first learned to be classified correctly but by the end of training is classified incorrectly. As a result, Toneva et al. (2019) established that there are two distinguished example classes: the first one get frequently forgotten, while the second one not at all; and the "unforgettable" examples can be excluded from the training set while still maintaining state-of-the-art generalization performance.

5.3.2 Error Analysis

I carried out a manual error analysis of the baseline system examining approximately 33% (200 out of 609) of the errors of a single run and classified it by (1) Question topic; (2) Question type; (3) Reasoning type, as described in Section 5.2.2; and, finally, (4) possible reason for the error. Table 5.2 contains detailed data for all four categorisations.

Question Topic			**Question Type**		
Category	**#**	**Percent**	**Category**	**#**	**Percent**
Nature / Science	47	23.5%	Definitional	45	22.5%
Law / Government	34	17.0%	Existence	43	21.5%
Sports	20	10.0%	Event Occurrence	22	11.0%
Entertainment / Media	33	16.5%	Possibility	23	11.5%
Fictional Events	9	4.5%	Other General Fact	26	13.0%
History	7	3.5%	Other Entity Fact	41	20.5%
Other	50	25.0%			
– Geo / Location	9	4.5%			
– Cooking / Food	14	7.0%			
Total	200	100%	Total	200	100%

Reasoning Type			**Reason for Error**		
Category	**#**	**Percent**	**Category**	**#**	**Percent**
Paraphrasing	97	48.5%	Errors	11	5.5 %
By Example	7	3.5 %	Other Data Issue	16	8 %
Factual Reasoning	12	6.0 %	Difficult/Confusing	19	9.5 %
Implicit	39	19.5%	Direct Understanding	103	51.5 %
Missing Mention	28	14.0%	Hierarchy	22	11 %
Other Inference	16	8.0%	Entity Property	18	9 %
			Timeline	11	5.5 %
Total	200	100%	Total	200	100%

Table 5.2: BoolQ error analysis categorised by topic (Geo/location and Cooking/Food categories are counted in the Other), question type, reasoning and possible error reason.

The majority of errors are related to *Definitional*, *Existence* and *Other Entity Fact* questions (22.5%, 21.5%, and 20.5 correspondingly). The rest of the question are relatively equally spread across the remaining categories. The absolute majority of errors belongs to *Paraphrasing* type questions (48.5%), which means the answer is actually in the passage and only a minimum amount of extra knowledge and reasoning, if any, may be required to answer the question. *Implicit* and *Missing Mention* account for 19.6% and 14.1% of errors respectively. 8% refer to *Other Inference*, 6% require *Factual Reasoning*, and only about 3.5% of errors require an understanding of provided examples.

Examining the cases which were not answered correctly by the $BERT$ baseline (one run) I tried to understand what could be the reason for the system failure. The most common reason (77%) is misunderstanding, which includes categories *Direct Understanding, Hierarchy, Entity Property*, and *Timeline* (see Table 5.2), which means the answer is in the passage but the system did not find/understand it. **Direct Understanding** is the simplest case where the passage contains the straightforward answer to the question. No extra world knowledge is involved in the reasoning process. Apart from direct understanding I also specify three additional sub classes: *Timeline*, *Entity Property*, and *Hierarchy*. Handling this phenomena might help to avoid a significant amount of errors.

- **Timeline** – the passage is organised as a timeline and it is important to understand not only if something happened or some statement is true or false but also *when* it happened or if it was true/false. For example, in (5.33), the passage talks about prices of bridge tolls and how it was organised but then it says "*tolls on the Port Mann Bridge were removed on September 1, 2017*" which leads to the answer "No".

  (5.33) **Question:** *is the port mann bridge a toll bridge*

  Passage: *In order to recover construction and operating costs,* ***the bridge was electronically tolled when originally built.*** *The toll rates increased to $1.60 for motorcycle, $3.15 for cars, $6.30 for small trucks and $9.45 for large trucks on August 15, 2015. Through increased prices and greater traffic, Transportation Investment Corporation (TI Corp), the public Crown corporation responsible for toll operations on the Port Mann Bridge, forecast its revenue would grow by 85% between fiscal years 2014 and 2017. ... Tolls were expected to be removed by the year of 2050 or after collecting $3.3 billion. As announced by B.C. Premier John Horgan a few days earlier,* ***all tolls on the Port Mann Bridge were removed on September 1, 2017****. Debt service was transferred to the province of British Columbia at a cost of $135 million per year.*

  Answer: No

- **Entity Property** the question is asking about a particular property of the object and some background knowledge about the object is required, or the question is about some quite specifically defined property and not much reasoning is needed. Or in another words, the question is about something that can be very easily structured. For example, in (5.34) it is necessary to understand that *now tv* is an internet television service and it may have a cartoon as an available media or not. And in Example (5.35) the World Cup championship is a very well defined concept and all countries who have ever won it are listed in the passage. In both these examples the answer "No" can be concluded based on *missing mention* type of reasoning. In both examples, questions can be answered only with an understanding of the entities, which are generally known (or can be easily obtained from additional sources) but not explicitly described in the passage.

(5.34) **Question:** *is how to train your dragon on now tv*

Passage: *The How to Train Your Dragon franchise from DreamWorks Animation consists of two feature films How to Train Your Dragon (2010) and How to Train Your Dragon 2 (2014), with a third feature film, How to Train Your Dragon: The Hidden World, set for a 2019 release. ... a new television series, titled Dragons: Race to the Edge, aired on Netflix in June 2015. The second season of the show was added to Netflix in January 2016 and a third season in June 2016. A fourth season aired on Netflix in February 2017, a fifth season in August 2017, and a sixth and final season on February 16, 2018.*

Answer: No

(5.35) **Question:** *has the u s ever won a world cup*

Passage: *The 21 World Cup tournaments have been won by eight national teams. Brazil have won five times, and they are the only team to have played in every tournament. The other World Cup winners are Germany and Italy, with four titles each; Argentina, France and inaugural winner Uruguay, with two titles each; and England and Spain with one title each.*

Answer: No

- **Hierarchy** the question is asking about something being a part of something else and the discussed concepts form a hierarchy, and/or an understanding of the hierarchy of concepts is necessary to answer the question. For example, in (5.36) the question is about a particular college belonging to a particular league.

  (5.36) **Question:** *is college of william and mary an ivy league school*

  Passage: *... Ivy League institutions account for seven of the nine Colonial Colleges chartered before the American Revolution; the other two are Rutgers University and the College of William&Mary.*

  Answer: Yes

27 out of 200 samples (13.5%), in my opinion, were incorrect, e.g. (5.10)-(5.12): 11 contained information in the passage which contradicted to the answer tag, and in another 16 cases, the passage did not contain enough information to conclude either a positive or negative answer. As I analysed only errors I was able to find false negative errors meaning the system provided a correct answer but it was counted as incorrect due to an incorrect answer label. I do not know how many false positive errors are in the data, where the system provided an incorrect answer which matched the answer label and was counted as correct.

19 samples I classified as difficult, including a few samples about particular relationships between fictional characters, and passages containing a long extract of text where it is quite hard to conclude the actual answer (see Example (5.3)). I also included in this category questions where domain knowledge is required to understand the passage and provide the answer. In contrast with previously described cases, where the average person would be able to provide the answer based on the passage, in this case only people who understand the field would be able to provide the answer based on the passage. It is very difficult to answer such questions.

I present the similarity comparison of percentages between data categorisation reported by Clark et al. (2019) and my error analysis in Figure 5.4. There is a similarity

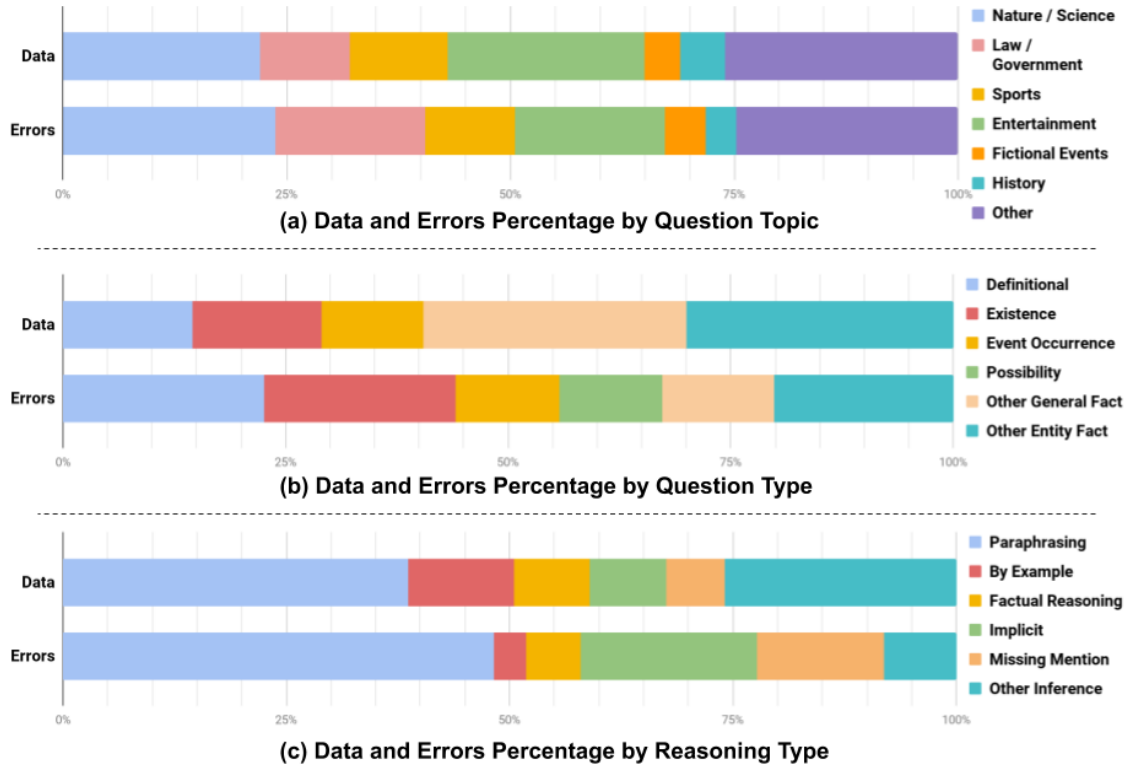


Figure 5.4: Similarity between data and errors in BoolQ dataset. (a) Data and errors percentage by question topic. (b) Data and errors percentage by question type. (c) Data and errors percentage by reasoning type.

between data and errors categorised by question topic (see Figure 5.4 (a)), and a slight similarity by question type (see Figure 5.4 (b)). The errors are split between question topics proportionally to the data reported by Clark et al. (2019). There are more errors in *Law/Government* questions and fewer errors in *Entertainment* questions. The majority of *Possibility* questions are coming from the *Other General Fact* category. There is no similarity between the data and error sample reasoning type distributions. As mentioned above, the majority of errors belong to the *Paraphrasing* (48.5%) category while in the data sample less than 40% of samples are this type. *Other Inference* accounts for more than 25% of the data but only 8% of the errors. Questions which can be answered *By Example* are 11.8% of the data sample but only 3.5% of the error sample.

5.4 Summary

In this chapter I discussed the task of boolean question answering. I examined the BoolQ dataset, carried out an the error analysis of the baseline system, introduced the measure of stable accuracy, and discovered the issue with the stability of results.

My main findings are:

1. A close look into the BoolQ dataset reveals that out of 200 samples from the $BERT_{large} + MNLI$ baseline errors almost 6% of samples contain an incorrect answer tag, and 8% do not contain enough evidence to answer the question.

2. The error analysis shows that 77% of errors are coming from rephrasing and are generally possible to answer.

3. The remaining 9% I classified as difficult questions as they involve deep understanding, reasoning, specific knowledge, and sometimes depend on opinion.

4. Although transformer-based systems (like $BERT_{large}$ and $RoBERTa$) demonstrate strong baseline results (82-85% accuracy) on the BoolQ dataset, the stable accuracy is statistically significantly lower. Concurring with McCoy et al. (2020), I conclude that it is important to consider multiple runs of a system, as models can significantly vary in their performance.

Chapter 6

Incorporating Knowledge Graphs into Boolean Question Answering

> Without a systematic way to start and keep data clean, bad data will happen.
>
> *Donato Diorio*

Back in Chapter 2 I described the task of boolean reading comprehension question answering and in the previous chapter I talked particularly about the BoolQ dataset (Clark et al., 2019).

Let's look back at Example 5.4 (repeated below as Example 6.1) where a question is asked about *Smeagol* but the passage is talking about *Gollum*, who turned out to be the same character. This question can be answered correctly only with the additional knowledge that *Smeagol* turned into *Gollum*. If such background information is not a part of the paragraph, exactly like in this example, the system could get this information from an additional source, such as a knowledge graph (KG).

(6.1) **Question:** *does smeagol die in lord of the rings*

Passage: *The Ring, which Gollum referred to as "my precious" or "precious", extended his life far beyond natural limits. Centuries of the Ring's influence twisted Gollum's body and mind, and, by the time of the novels, he "loved and hated (the*

Ring), just as he loved and hated himself." Throughout the story, Gollum was torn between his lust for the Ring and his desire to be free of it. Bilbo Baggins found the Ring and took it for his own, and Gollum afterwards pursued it for the rest of his life. Gollum finally seized the Ring from Frodo Baggins at the Cracks of Doom in Orodruin in Mordor; ***but he fell into the fires of the volcano, where both he and the Ring were destroyed.***

Answer: Yes

Based on the error analysis described in Chapter 5 I established that approximately 20% of errors are related to a particular property or position in a hierarchy of objects/concepts. In other words, the usage of structural information about objects or concepts can help in answering those questions.

The usage of KGs seems like a promising intuitive solution for answering a wide range of questions. This chapter describes experiments which combine KGs with reading comprehension models. I investigate two research questions:

- Can a reading comprehension model benefit from adding textual data obtained from knowledge base?
- Can adding structured information about entities in the question and passage, and the relations between them help to answer questions in a reading comprehension setting?

The majority of the experiments described in this chapter was carried out during an internship at Google Research Switzerland between September and December 2019 in collaboration with Massimo Nicosia.

The chapter is organised as follows: first, in Section 6.1 I discuss available knowledge graphs and their use in question answering. In Section 6.2 I describe two approaches to using Knowledge Graphs in reading comprehension: adding an extra sentence to the passage, and using structured data based on a Graph Neural Network (GNN) architecture. Finally, I draw some conclusions in Section 6.3

6.1 Knowledge Graphs

Knowledge Graphs are a popular technology used by many search engines, knowledge-engines, and question-answering services. The original idea was introduced by Schneider (1973). The concept has been defined and interpreted in multiple ways (Pujara et al., 2013; Paulheim, 2017; Färber et al., 2017; Ehrlinger and Wöß, 2016). I define knowledge graph as follows:

> **Definition 6.1.** A knowledge graph (KG) is a graph-structured data model which integrates knowledge and data consisting of entities, their semantic types, properties, and relationships between entities.

One of the first knowledge graphs was WordNet[1] proposed by Fellbaum and Miller (1998), and also some topic-specific KGs, e.g. Geonames.[2] Later more general-purpose knowledge bases started to appear such as DBPedia,[3] (Auer et al., 2007), Freebase,[4] (Bollacker et al., 2008), and YAGO[5] (Suchanek et al., 2007). Nowadays, a number of multi-national companies use their own knowledge graphs, such as Facebook Graph Search;[6] AirBnB,[7] The LinkedIn Knowledge Graph[8] (He et al., 2020). In this work I look into two particular knowledge graphs: ConceptNet[9] (Liu and Singh, 2004; Speer et al., 2017) and Google Knowledge Graph.[10]

[1] `https://wordnet.princeton.edu/` – last verified May 2021

[2] `https://www.geonames.org/` – last verified May 2021

[3] `https://wiki.dbpedia.org/` – last verified May 2021

[4] `https://en.wikipedia.org/wiki/Freebase_(database)` – last verified May 2021. I provide the Wikipedia link as Freebase.com was officially shut down in 2016.

[5] `https://yago-knowledge.org/`– last verified May 2021

[6] Facebook Graph Search was almost entirely deprecated in June 2019

[7] `https://airbnb.io/projects/knowledge-repo/` – last verified May 2021

[8] `https://engineering.linkedin.com/blog/2016/10/building-the-linkedin-\knowledge-graph` – last verified May 2021

[9] `https://conceptnet.io/` – last verified May 2021

[10] `https://googleblog.blogspot.com/2012/05/introducing-knowledge-\graph-things-not.html` – last verified May 2021

6.1.1 ConceptNet

ConceptNet is an open semantic network first introduced by Liu and Singh (2004). It is based on DBPedia, Wiktionary,[11] Open Multilingual WordNet[12] (Bond and Foster, 2013), OpenCyc,[13] and other resources, created for computers to understand the words and concepts in the same way people do. In this work I use the ConceptNet version 5.5 described by Speer et al. (2017). It contains over 21 million edges and over 8 million nodes (concepts) in over 80 languages, with 1.5 million nodes in English. It was particularly designed to be used in NLP, for example, in word embeddings.

ConceptNet captures common-sense knowledge where words and phrases of natural language are presented as entities with labeled edges. According to the documentation,[14] ConceptNet has 34 relation types such as `''PartOf''`, `''HasProperty''`, `''Synonym''`, and others. There are a few more relations like `''language''` or `''occupations''` which are not mentioned in the documentation.

A clear REST API[15] is available and makes ConceptNet easy to use. It returns data in JSON-LD format, containing start/end entities, relation type, language and other meta information. Some relations contain a `surfaceText` which represent start/end entities and relation as natural language text. For example for the entity *"polar bear"* there is a relation `''Synonym''` to the entity *"ice bear"*, and the `surfaceText` *"[[polar bear]] is a synonym of [[ice bear]]"* is provided. Multi-word entities can be searched by replacing the space with the "_" symbol.

6.1.2 Google Knowledge Graph

Back in 2012, Google introduced the Google Knowledge Graph based on DBpedia and Freebase. The goal of the Google Knowledge Graph is to improve the quality of search

[11] `https://www.wiktionary.org/` – last verified May 2021
[12] `http://compling.hss.ntu.edu.sg/omw/` – last verified May 2021
[13] `http://www.qrg.northwestern.edu/OpenCyc/index_opencyc.html` – last verified May 2021
[14] `https://github.com/commonsense/conceptnet5/wiki/Relations` – last verified May 2021
[15] `api.conceptnet.io` – last verified May 2021

results and provide structured information in the form of an info box (the structured information in the info box was presented in Figures 5.1 and 5.2 (a) in Chapter 5). It has more than 500 billion facts about 5 billion entities.[16] The entities describe real-world objects and concepts like people, places, events, and things.

The Google Knowledge Graph also has an API available by authorized requests. This way a JSON-LD result about particular entity can be obtained and used as an additional structured information for answering questions.

6.1.3 Using Knowledge Graphs in Question Answering

Knowledge Graphs have been widely used in the Natural Language Processing and there is an entire area of NLP devoted to answering questions over KG (Berant et al., 2013; Mohammed et al., 2018; Lukovnikov et al., 2017; Zhang et al., 2018c). Mostly these are factoid questions and question which can be formulated as queries.

Some applications of KG have been used in the reading comprehension task as well. A number of datasets mentioned in Chapter 2 (Qangaroo-MedHop, WikiMovies, and Wikireading) were created with usage of such KGs as WIKIDATA,[17] DRUGBANK,[18] OMDb,[19] while the CommonsenceQA dataset (Talmor et al., 2019) was created using ConceptNet. Weissenborn et al. (2017); Bauer et al. (2018); Mihaylov and Frank (2018); Lin et al. (2019); Qiu et al. (2019b) also use ConceptNet as a source of external knowledge. They archived state-of-the-art performance for both entity extraction and event extraction on the ACE 2005 Multilingual Training Corpus[20] (Christopher Walker, 2006).

Yang and Mitchell (2017) investigate the approach of adding external Knowledge Graph information for recurrent neural networks (LSTM) for machine reading.

Mihaylov and Frank (2018) use common sense knowledge from the Open Mind Common Sense[21] (Singh et al., 2002) part of ConceptNet and an attention mechanism to re-

[16] `https://blog.google/products/search/about-knowledge-graph-and-\knowledge-panels/` – last verified May 2021

[17] `https://www.wikidata.org/wiki/Wikidata:Main_Page` – last verified May 2021

[18] `https://www.drugbank.ca/` – last verified May 2021

[19] `http://www.omdbapi.com/` – last verified May 2021

[20] `https://catalog.ldc.upenn.edu/LDC2006T06` – last verified June 2021

[21] `https://www.media.mit.edu/projects/open-mind-common-sense/overview/`

trieve relevant pieces of knowledge for cloze reading comprehension task, evaluated on Children's Book Test (Hill et al., 2016).

Qiu et al. (2019b) introduced the Structural Knowledge Graph-aware Network (SKG) model which constructs a sub-graph from a knowledge graph based on the context and is capable of dynamically updating the representation of the knowledge according to the structural information of the sub-graph. The authors evaluate the method on the ReCoRD (Zhang et al., 2018a) dataset and achieve state-of-the-art performance.[22]

Weissenborn et al. (2017) use ConceptNet and Wikipedia, transform additional knowledge into free text format, and evaluate on SQuAD (Rajpurkar et al., 2016) and TriviaQA (Joshi et al., 2017). The authors contextually refine word embedding by incorporating the data into the model, so after reading the passage and question, the embeddings are updated according to assertions extracted from KG.

Bauer et al. (2018) use pointwise mutual information and term-frequency based scoring function to select common-sense information from ConceptNet and, with a selectively gated attention mechanism over a path of grounded multi-hop relational knowledge, improve multi-hop reasoning. The authors evaluate on NarrativeQA (Kočiský et al., 2018) and QAngaroo-WikiHop (Welbl et al., 2018).

Yang et al. (2019) investigate the possibility of combining BERT with external KGs. They introduce KT-NET which first passes the question+passage input into BERT, then integrate external knowledge by enriching BERT representation with context-aware and knowledge-aware KB embeddings, then use attention to select the most relevant KG concepts. They use WordNet and NELL (Carlson et al., 2010) as knowledge graphs, and evaluate on ReCoRD (Zhang et al., 2018a) and SQuAD (Rajpurkar et al., 2016).

– last verified June 2021

[22] According to the leader board: `https://sheng-z.github.io/ReCoRD-explorer/` the method was at top 3 at the time of submission and currently shares fourth and fifth place – last verified May 2021

6.2 Knowledge Graphs for Answering BoolQ Questions

Before using any sort of KG it is necessary to apply entity recognition and linking. To extract entities I used the CloudAPI.[23] It brings structure to the text: tokenization; part of speech tagging; named and unnamed entities including the Freebase[24] KG identifier (MID); mark measures such as numbers, dates etc; and VerbNet[25] annotations which is used for establishing a relation between entities. An example of CloudAPI output[26] is presented in 6.2.

(6.2) **Input sentence:** Michelangelo Caravaggio, Italian painter, is known for "The Calling of Saint Matthew"
CloudAPI output:


```
{"entities": [
  { "name": "Michelangelo Caravaggio",
    "type": "PERSON",
    "metadata": {
      "wikipedia_url": "http://en.wikipedia.org/wiki/
         Caravaggio",
      "mid": "/m/020bg"  },
    "salience": 0.83047235,
    "mentions": [
      { "text": {
          "content": "Michelangelo Caravaggio",
          "beginOffset": 0 },
        "type": "PROPER" },
      { "text": {
          "content": "painter",
          "beginOffset": 33 },
        "type": "COMMON" } ] },
  { "name": "Italian",
    "type": "LOCATION",
    "metadata": {
      "mid": "/m/03rjj",
      "wikipedia_url": "http://en.wikipedia.org/wiki/
         Italy" },
    "salience": 0.13870546,
    "mentions": [
```


[23] https://cloud.google.com/apis/docs/overview – last verified May 2021
[24] https://en.wikipedia.org/wiki/Freebase_(database) – last verified May 2021
[25] http://verbs.colorado.edu/~mpalmer/projects/verbnet.html – last verified May 2021
[26] The example is taken from https://cloud.google.com/natural-language/docs/quickstart – last verified May 2021

```
24          { "text": {
25              "content": "Italian",
26              "beginOffset": 25 },
27            "type": "PROPER" } ] },
28      { "name": "The Calling of Saint Matthew",
29        "type": "EVENT",
30        "metadata": {
31          "mid": "/m/085_p7",
32          "wikipedia_url": "http://en.wikipedia.org/wiki/
               The_Calling_of_St_Matthew_(Caravaggio)" },
33        "salience": 0.030822212,
34        "mentions": [
35          { "text": {
36              "content": "The Calling of Saint Matthew",
37              "beginOffset": 69 },
38            "type": "PROPER" } ] }
39    ],
40    "language": "en"
41  }
```

The sentence has three entities recognised by CloudAPI: *"Michelangelo Caravaggio"*, *"Italian"*, and *"The Calling of Saint Matthew"* (see lines 2, 17 and 28 in (6.2)). Each entity has a `type` such as PERSON, LOCATION, or EVENT, `metadata` including Wikipedia URL and Freebase identifier (MID), and `mentions` which show the text and connections, e.g. the first entity is mentioned twice: first as a name of the person starting at position 0, and then as *painter* starting at position 33 (see lines 8-16 in (6.2)).

6.2.1 Knowledge Graph Text Extension

Partly inspired by the work of Weissenborn et al. (2017) I apply a straightforward approach to converting relations extracted from ConceptNet into text, add the text to the passage, and run the baseline BERT model.

For every entity recognised by the entity linker I build a request, extract the top 100 relations, and filter out all non-English ones. I removed unrelated relations such as `''External URLs''` because it does not contain any additional information which might help in answering the question, and `''FormOf''` because it is too general and contains a lot of repetitions. I transform ConceptNet relations into simple sentences based

on the relation description (the *surfaceText*) or, if there is no description, I created a string: `[entity1] [relation] [entity2]`. Figure 6.1 shows an example of the usage of how ConceptNet entities are used to answer the question *"is anne with an e filmed on pei"* from Example (6.3). The entity *"pei"* is actually a synonym of *"Prince Edward Island"*. This connection is missing from the context and can be obtained from the KG by adding the sentence *"pei is a synonym of Prince Edward Island"*.

(6.3) **Question:** *is anne with an e filmed on pei*

Passage: *The series is filmed partially in Prince Edward Island ...*

Answer: Yes

Figure 6.1: The example of usage ConceptNet entities for answering a Boolean question.

6.2.1.1 Sentence Extraction and Filtering

Since a long input might confuse the model, I add extra sentences only to the passages where they can better "explain" the nature of entities. To select those sentences, I rank all

extracted sentences S according to the sum of their similarities with the question q and passage p:

$$score(s) = g(k(s), k(q)) + g(k(s), k(p))$$

where $g \in \{correlation, cosine\}$ similarities, k is a semantic embedding. I used the semantic textual similarity model proposed by Yang et al. (2018a) (see Section 3.1.3 in Chapter 3) and available via TensorFlowHub[27] (Cer et al., 2018). To filter out more examples, I added an empirically obtained threshold for similarities: $correlation > 220$, and *cosine similarity* > 1.38. I add a sentence to the passage only if the former was ranked as the most similar to the question and passage by both correlation (inner product) and cosine similarity, and each score is higher than the established thresholds (`SentEmb`). In this manner I add sentences to 21.84% of passages.

Another method of selecting relevant sentences is to look only at those relations which connect entities from the question with entities from the passage (`Q&P Match`). This way, 22.58% of data is affected. Then I combined these two strategies adding sentences only to those examples which meet both criteria (`Intersect`) and all those where at least one of the criteria is met (`Union`).

6.2.1.2 Results

All experiments in Table 6.1 are reported based on 5 runs for each sentence extraction setting, except the `Intersect` which was run 5 times twice (that is why there is an extra column in the table).

The baseline ($BERT + MNLI$) is described in detail in the previous chapter and gives the average accuracy of 81.26% and stable accuracy if 73.84%.

Adding the sentences selected by thresholds, I obtain an average accuracy of 81.23% (see Table 6.1: `SentEmb`). Connecting only entities from the question with entities from passage gives a slightly worse performance (Table 6.1: `Q&P Match`). The intersection

[27] `https://tfhub.dev/google/universal-sentence-encoder-qa-large/1` – last verified November 2021
Here I use the term correlation as defined in TensorFlowHub: correlation is calculated as an inner product of question and answer embeddings.

gives the best performance. By affecting only 1.23% of data I obtained 81.46% accuracy as an average of 5 runs and 82.05% accuracy for ensemble majority voting scenario (Table 6.1: `Intersect`). `Union` did not show any improvement on accuracy.[28]

	Baseline	**Sent Emb**	**Q&P Match**	**Intersect**	**Union**
Data Coverage (%)	-	21.84	22.58	1.23	38.57
AVG Accuracy	**81.26**	81.23	80.86	**81.46**	80.72
Stable Accuracy	73.84	73.19	72.61	73.74	72.40
Majority voting	81.62	81.89	81.37	**82.05**	81.10

Table 6.1: Percentage of data changed and accuracy over 5 runs: average (**AVG**), stable (**Stable**), and majority voting (**Majority**).

6.2.1.3 Analysis

Table 6.2 presents the number of new stable correct and new fluctuating samples compared to the baseline. Looking into particular samples, I observe a positive tendency towards stable correct answers. For `SentEmb` and `Union` the number of new consistently correctly answered questions is higher than new errors. For `SentEmb`, `Q&PMatch`, and `Union` the number of questions where the predicted answer fluctuates from error is also higher than the number of questions where the predicted answer fluctuates from correct. The `Intersect` approach shows minor improvement as there are only 4 new stable correctly answered questions and no new stable errors but 6 more questions where the answer becomes fluctuated (one from error to correct and 5 from correct to error).

New		**Sent Emb**	**Q&P Match**	**Inter section**	**Union**
Stable	**Correct**	27	27	4	54
	Error	18	28	0	32
Fluctuating	**Error→Correct**	34	44	1	65
	Correct→Error	19	23	5	42

Table 6.2: **New Correct (Error)** corresponds to the number of new stable (wrt to baseline) correct (incorrect) predictions, **New Fluctuating** is the number of new questions where the answer fluctuates: **Error→Correct** (**Correct→Error**) is the number of questions where the answer was a stable error (correct), becoming correct (error) sometimes.

[28] According to the two sample proportion Z-Test the maximum difference: $z = -1.3674$, $p = 0.17068$

6.2.2 Modeling Knowledge Graph with GraphNNs

Another approach I tried is to change the architecture. I adopted the model proposed by Vaswani et al. (2017) and modified by Shaw et al. (2019). It is based on transformer and Graph Neural Network (GNN) and created to generate logical forms. First, entities, relations, and input tokens are encoded, then they are passed into a GNN sub-layer that incorporates edge representations (which is an extension of the self-attention sub-layer, see 3.4.5.2 in Chapter 3 for a detailed explanation). Then a Feed-Forward Network is applied. Shaw et al. (2019) also extend the decoder with a action selection and copy mechanism (Vinyals et al., 2015) which I do not use, as the model works for us as a sequence-to-sequence model but generates the answer "Yes" or "No" instead of a logical form (see Figure 6.2).[29]

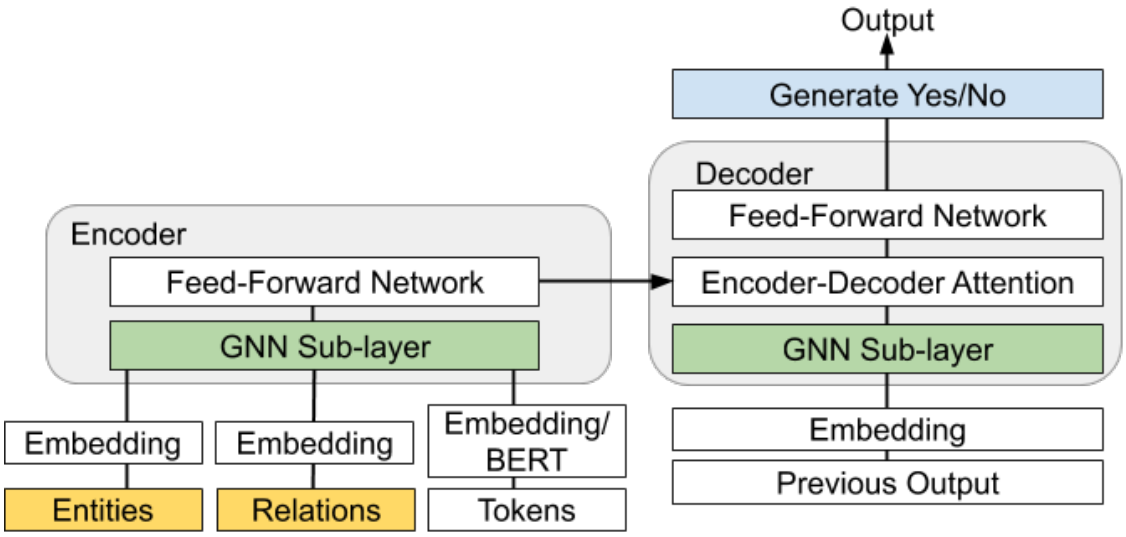


Figure 6.2: The used system GNN architecture based on Shaw et al. (2019) approach without action selection and copy mechanism.

Now, instead of ConceptNet I used the Google Knowledge Graph to obtain entity relations. Entities are represented as semantic nodes and they are connected if there is a relation between the corresponding entities. The relations can be simple, meaning it just indicates there is a relation, or contain a type indicating what kind of relation is there. There are a number of parameters (flags) that could be explored:

1. add a relation between different mentions of the same entity;

[29] I do not use a binary classification here as I was adapting the existing system.

2. add a relation between different entities which have the same MID;

3. consider only connections between entities from question and passage but not connections between entities only from question (or passage), as was done in the ConceptNet `Q&AMatch` experiment;

4. distinguish different types of relations;

5. add relations to entities not mentioned in the question or passage which connect to another mentioned entity.

To apply transfer learning and pre-train the model on other datasets, the data should be annotated with dependencies as described above. To the best of my knowledge, there is no suitable data with such dependencies available, so I compare the results with a pre-trained $BERT_{large}$ model without the usage of MNLI data and fine tune only on BoolQ.

The results are presented in Table 6.3. The first row shows the baseline $BERT$ model with no KG data and the remaining rows show the $BERT+GNN$ system (1) with no KG data, (2) with ConceptNet, or (3) with the Google Knowledge Graph. I also experimented with adding relations between entities with the same MID, adding relations types, and relations only between entities from question and passage (points 2-4 mentioned above). Non of those methods outperform the baseline. None of the differences between the baseline are statistically significant (according to the two Sample proportion Z-Test the maximum difference: $z = -1.3674$, $p = 0.17068$).

	No KG	**+ConceptNet**	**+GKG**
$BERT_{large}$ (baseline)	**78.09***	-	-
$GNN + BERT$	77.37*	77.4	76.80
+ Same MID	-	-	77.60
+ Relation Type	-	-	77.75
+ Q&AMatch	-	-	76.95

Table 6.3: Preliminary accuracy results (single run) on development set for Graph Neuron Network (**GNN**) with usage of ConceptNet and Google KG (**GKG**). Where * indicates the parameters of learning were properly tuned.

6.2.3 Observations

6.2.3.1 Entity Recognition and Linking

Working on the experiments described above I observed a problem with the entity linker. Entity recognition and linking works very well for passages where data is well formatted and the original Wikipedia case is kept. But for lower cased questions it does not work the way it was expected. Named entities are often missed or the MID is missing. In some extreme cases the entity is recognised but the wrong MID was assigned. For example, in the question *"is **northern ireland** part of the **great britain**"* the entity "*northern ireland*" is not recognised but the entity "*ireland*" (Republic of Ireland) is mentioned instead. The entity "*great britain*" is recognised with the MID of United Kingdom, although they are not the same (see Figure 6.3). This is an illustrative example of a question which was answered incorrectly by all runs of the baseline system. Adding KG information should help as it defines entities and relationships between them in exactly the way the question is asking for but due to the inaccuracy of entity recognition and linking, the additional data does not help.

6.2.3.2 Missing Entities in ConceptNet

Frequently the relations from the ConceptNet are too general and do not bring any new information which might be already known by language model, e.g. *"cookie jar is a type of jar"*.[30] In contrast, there is also a lack of information for specific entities. Entities that do not exist in the dataset or relations are completely useless for the BoolQ setting, e.g. there is an entity *"Tom Hanks"*,[31] but no such entity as *"Meg Ryan"*,[32] or the entity *"dragon ball"* contains only non-English connections.[33]

[30] `https://conceptnet.io/c/en/jar` – last verified May 2021

[31] `https://conceptnet.io/c/en/tom_hanks` – An English term in ConceptNet 5.8 – last verified May 2021

[32] `https://conceptnet.io/c/en/meg_ryan` – 'meg ryan' is not a node in ConceptNet. – last verified May 2021

[33] `https://conceptnet.io/c/en/dragon_ball` – last verified May 2021

(a) Difference between United Kingdom and Great Britain

Question: *is* ***northern ireland*** */m/05bcl* *part of the* ***great britain*** */m/034cm*

Passage: *The island is dominated by a maritime climate with quite narrow temperature differences between seasons. Politically,* ***Great Britain*** */m/034cm* *is part of* ***the United Kingdom of Great Britain and Northern Ireland***, */m/07ssc* *and constitutes most of its territory. Most of England, Scotland, and Wales are on the island. The term ``Great Britain'' is often used to include the whole of England, Scotland and Wales including their component adjoining islands; and is also occasionally but contentiously applied to the UK as a whole in some contexts.*

Answer: *No (false)*

(b) Gold KG annotation

Question: *is* ***northern ireland*** */m/03rt9* *part of the* ***great britain*** */m/07ssc*

(c) CloudAPI KG annotation

Figure 6.3: The example of difference between expected and actual annotations from entity linker. (a) The map illustrates the difference between the United Kingdom and Great Britain (The map is created by Anna Debenham and taken from `https://brilliantmaps.com/eng-gb-uk/` – last verified May 2021). Expected (b) and received (c) entity annotations and MID from Google Knowledge Graph. Where:
m/05bcl – Northern Ireland `https://freebase.toolforge.org/freebase/m/05bcl`
m/03rt9 – Ireland (Republic) `https://freebase.toolforge.org/freebase/m/03rt9`
m/034cm – Great Britain `https://freebase.toolforge.org/freebase/m/034cm`
m/07ssc – the United Kingdom of Great Britain and Northern Ireland `https://freebase.toolforge.org/freebase/m/07ssc`; all links last verified May 2021.

6.2.3.3 Is a Knowledge Graph necessary?

The BoolQ dataset was not originally created to be used with knowledge graphs. The task implies that all related information required to answer a question is provided by the passage. For some questions, such as Example (6.3), the additional information is extremely helpful. For questions such as the one about Northern Ireland in Figure 6.3, even though the passage has all the required information, the correct by structured data from a KG could highlight the relation between entities and help answer the question. There are some cases where KG information is not needed or just cannot be applied, e.g. (6.4) and (6.5).

(6.4) **Question:** *do all ni numbers have a letter at the end*

Passage: *The format of the number is two prefix letters, six digits, and one suffix*

letter. The example used is typically QQ123456C. ...

Answer: Yes

(6.5) **Question:** *was the movie insomnia based on a book*

Passage: *Robert Westbrook adapted the screenplay to novel form, which was published by Alex in May 2002.*

Answer: No

In Example (6.4) a question is asked about a number format and the information about the specific last symbol is unlikely to be a part of a KG. Example (6.5) contains a very short passage explicitly saying there is a book but it was adapted from the screenplay. In this case, a KG could provide potentially confusing information simply stating that there is a book.

6.3 Summary

In this chapter I tried two approaches to integrating knowledge graph information, one based on augmenting the passage text and another using a Graph Neural Network. Neither are successful. I encountered the following problems:

1. Automatic entity recognition and linking tend to miss and mismatch entities in BoolQ questions which are provided in lower case;

2. ConceptNet contains information which is very general and might be already in the $BERT$ language model (Petroni et al., 2019) show that $BERT$ contains relational knowledge and has a strong ability to recall factual knowledge without fine-tuning. Adding ConceptNet information as additional sentences into a passage might slightly help the performance but no clear improvement is visible;

3. As I observe a positive tendency towards stable correct answers in the ConceptNet experiments, I conclude the graph based neural network can be used for answering

Boolean questions but, possibly due to inaccuracy in the entity linking and KG, the performance of this method does not exceed the baseline performance;

4. I also suggest that the number of questions where suitable KG data is needed and could be found might just not be enough for the models to learn from.

The result of this work has been published (Dzendzik et al., 2020) on Insights from Negative Results Workshop[34] (co-located with EMNLP 2020).

[34] https://aclanthology.org/2020.insights-1.2/ – last verified November 2021

Chapter 7

Answering User-Generated Questions

> My advice is to answer every customer, in every channel, every time. This is different from how most businesses interact with customers, especially online, which is to answer some complaints, in some channels, some of the time.
>
> *Jay Baer*

It has become commonplace for people to share their opinions about all kinds of products by posting reviews online. These questions can take the form of search engine queries or questions posted to dedicated fora. It has also become commonplace for potential customers to do research about the quality and limitations of these products by posting questions online. In this chapter I introduce the AmazonYesNo dataset based on user generated questions, answers, and passages (reviews) about products listed on the Amazon.com website[1]. No crowdsourcing has been involved in the creation of this dataset. I test the extent to which reviews are useful in question-answering by combining two Amazon datasets, and focusing on boolean questions.

The chapter is organised as follows: I start with related work on spontaneously user-generated content for question answering and reading comprehension in Section 7.1. In

[1] `www.amazon.com` – last verified May 2021

Section 7.2 I introduce the AmazonYesNo dataset. In Section 7.3 I look in detail at the parallel work of Gupta et al. (2019b), and in Section 7.4 I reconsider the data in light of it. Section 7.5 proposes an approach for answering the user generated questions, explains the experiment setup, and presents the results, followed by some error analysis in Section 7.6. Finally, I summarize the chapter in Section 7.7.

7.1 Related Work

A number of studies have explored the use of customer reviews in information retrieval and question answering.

I construct the AmazonYesNo dataset from two Amazon-based datasets: the first one is a collection of product reviews from 24 different domains[2] collected by He and McAuley (2016) and updated by Ni et al. (2019). The data contains information about products including Amazon Standard Identification Number (ASIN), reviewers, ratings and additional metadata such as price, sales-rank, brand info, and co-purchasing links. The second (Wan and McAuley, 2016; McAuley and Yang, 2016) contains a wide variety of questions and answers about products also with ASIN from 21 domains.

McAuley and Yang (2016) classified all questions collected from Amazon website into boolean and open-ended. For boolean questions the authors classified each answer with an answer tag, *Yes* or *No*, using a number of grammar rules provided by He and Dai (2011). The estimated accuracy of this approach is 88.3%.

Using this Amazon data, Yu et al. (2018) developed a framework which returns a ranked list of sentences from reviews or existing question-answer pairs for a given question. Xu et al. (2019) created a different dataset comprising Amazon laptop reviews and questions, and Yelp restaurant reviews and questions, where reviews are used to answer questions in multiple-turn dialogue form. Ni et al. (2019) use the review data from Yelp[3] and Amazon Clothing to address the recommendation justification task.[4] Bjerva et al.

[2]More details here: `http://jmcauley.ucsd.edu/data/amazon/` – last verified May 2021

[3]`https://www.yelp.com/dataset` – last verified May 2021

[4]According to Ni et al. (2019) the justifications task is to generate justifications of explanation as to why a recommendation might match a user's interests, that are relevant to users' decision-making process.

(2020) focused on subjective questions from TripAdvisor, Yelp (restaurants), and Amazon reviews (movies, books, electronics, and grocery). They created a new dataset called SubjQA.[5] Bogdanova et al. (2017) and Bogdanova and Foster (2016) do not use review data but also focus on QA over user-generated content, attempting to find similar questions or rank answers in user fora. The very recent work of Roy et al. (2020) applied large transformer models (XNet and BERT) to the 6 biggest domains of products from the Flipcart.com[6] website to answer user questions based on product specifications.

Parallel to this work, Gupta et al. (2019b) created the AmazonQA dataset from the same data as I did, i.e. collection of Amazon questions and reviews. As this work is extremely similar to mine and has been done at approximately the same time. I will provide more details about it in Section 7.3.

7.2 AmazonYesNo Data

The Amazon product and review datasets (He and McAuley, 2016; Ni et al., 2019) have 17 domains in common and products can be matched using the ASIN. I compile a dataset of questions about Amazon products together with consumer reviews of the same products using the Amazon data collection.

In order to obtain data with reviews, questions and answers, I first select all those products which contain reviews and questions, focusing on yes/no questions. I observe that the majority of questions can be answered "Yes" (65-75% depending on the domain), so I balance the data by selecting an equal amount of yes/no questions. This result is 80391 questions about 40806 products – see Table 7.1 for more details.

All data is fully user-generated except the answer tags which are provided by McAuley and Yang (2016). I use the subset of data with one answer per question and subset with contains 5 reviews per product.[7] An example of the combined data is shown below in (7.1) (I keep the original spelling).

[5]The SubjQA dataset is not mentioned in Chapter 2 as it has been introduced after the dataset collection was finished.

[6]`https://www.flipkart.com/` – last verified May 2021

[7]This is one of versions of data provided.

Domain	# Prdct	Question			Review		
		#	# S	# W	#	# S	# W
Automotive	574	1113	1469	14158	7276	34112	618k
Baby	1105	2163	2793	26513	48835	281953	5083k
Beauty	1522	2763	3537	29105	39381	205000	3437k
Cell Phones & Accessories	2401	5711	6836	60946	72407	369241	6836k
Clothing Shoes & Jewellery	251	479	622	5166	4349	19815	310k
Electronics	13683	27877	35073	330340	691400	4130768	78242k
Grocery & Gourmet Food	758	1223	1549	12288	17436	85097	1417k
Health & Personal Care	3259	5833	7491	63520	93189	488411	8658k
Home & Kitchen	6527	12003	15580	138021	215194	1230269	21313k
Musical Instruments	227	399	505	4642	3150	15642	284k
Office Products	624	1269	1574	14047	10200	79444	1598k
Patio Lawn & Garden	352	637	851	7935	4576	35604	712k
Pet Supplies	1132	1945	2722	25428	37538	202237	3574k
Sports & Outdoors	3455	6699	8366	75405	90501	452578	7958k
Tools & Home Improvement	2619	5245	6883	65978	47491	270010	4983k
Toys & Games	1719	3205	3975	34301	39456	215718	3712k
Video Games	598	1827	2192	19902	32790	291642	6071k
Total	40806	80391	102018	927695	1455169	8407541	154m

Table 7.1: Balanced AmazonYesNo dataset (v.1) statistics per domain: Number of products (**Prdct**) which have yes/no questions, number of questions (**# Question**), count of sentences in questions (**S**), total number of words in questions (**W**), total number of reviews (**# Reviews**), number of sentences in reviews, total number of words in reviews.

(7.1) **Product ASIN:** *B002PEGT9U* **Domain:** *Toys and Games*

Reviews: *...I was a little surprised at how much time it took to assemble. There were alot of the smaller parts that I would have assumed pre-assembled that weren't...*

Question: *Does it come assembled*

Answer: *No, count on, at least an hour to assemble.*

Answer Tag: *No*

The authentic user-generated nature of this dataset makes it significantly different from other reading comprehension datasets. The average length of questions is 11.5 tokens which is not that different from the majority of other RC datasets, but the number of instances (e.g. movie plots for MovieQA, Wikipedia article for SQuAD, and products for Amazon) is significantly bigger.

7.2.1 Question Distribution

To better understand the nature of questions I carried out some additional analysis looking into question formulation. 83% of questions contain a question mark (78–95% depending

on domain), 21% of questions are formulated with more than one sentence (16–31% depending on the domain), more than 25% of questions start with the word *Does*, and more than 15% with *Can*, *Is* or *Will*. Figure 7.1 shows the distribution of most frequent first words and two first words in the questions of AmazonYesNo dataset. Figure 7.2 shows examples of the questions for the most frequent two word question combinations in the dataset.

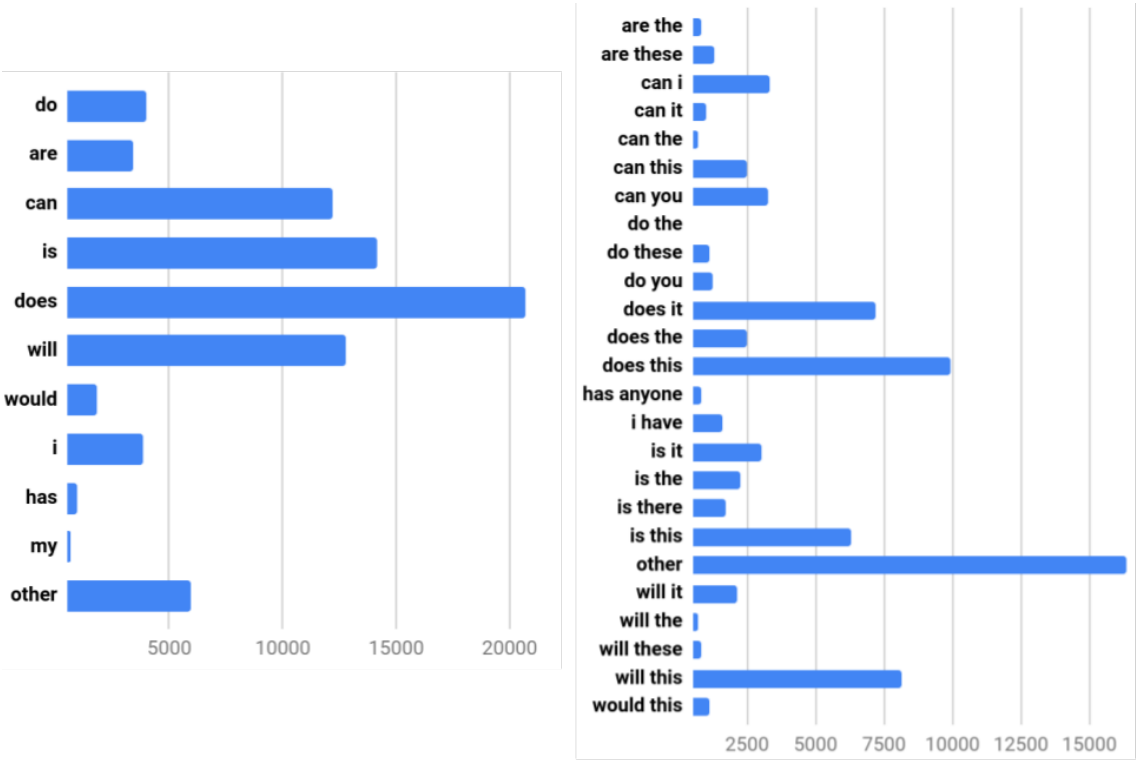


Figure 7.1: Distribution of the most frequent first words and two first words in the questions of AmazonYesNo dataset.

7.2.2 Type of Questions

According to Kaushik and Lipton (2018), Dunietz et al. (2020), and Rogers et al. (2020b), the difficulty of reading comprehension datasets has not been considered enough. I conduct a manual analysis to better understand the relationship between questions and reviews, to assess the feasibility of using user reviews to answer user questions and to estimate an upper bound on system performance. 100 questions from four domains are analysed. I categorise the questions into the following classes:

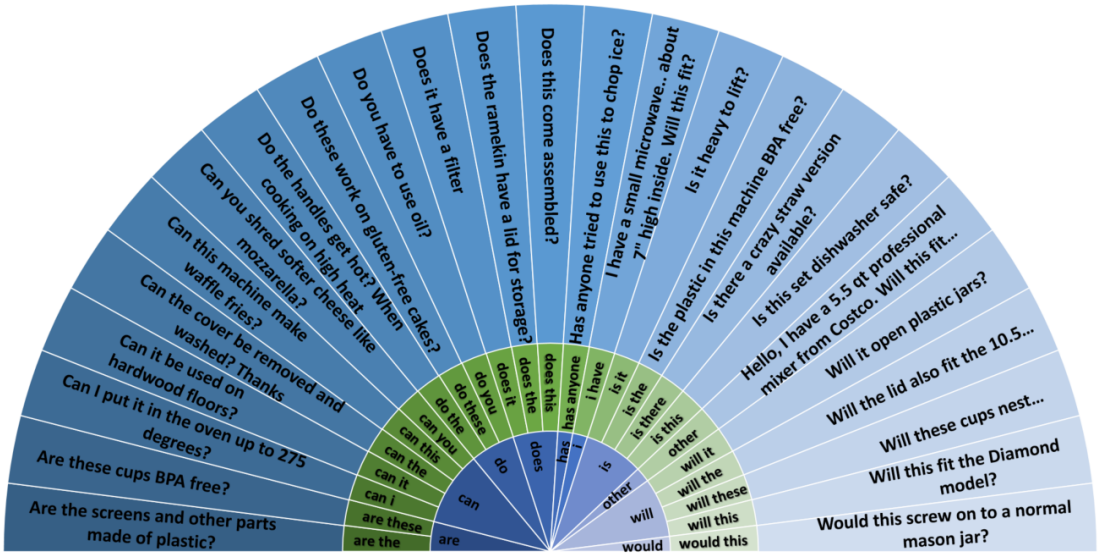


Figure 7.2: Examples of the questions for the most frequent first two words in AmazonYesNo dataset.

- **Straightforward:** the question is clearly and directly answered in the reviews, no extra world knowledge is needed e.g.

  (7.2) **Review:** *... I used two of these, one for each side of the bed.*

  Question: *can this product be used if 2 bed rails are needed for one bed?*

  Answer Tag: *Yes*

- **Indirect:** the question can be indirectly answered by the review, i.e. some extra world knowledge is required or, for example, some arithmetical comparison like in (7.3):

  (7.3) **Review:** *... it doesn't give an exact voltage and maxes out at 12.7 volts*

  Question: *Can this be used to charge a 48v battery?*

  Answer Tag: *No*

  Question: *Is this a good charger/jump starter for a 12v deep cell battery?*

  Answer Tag: *Yes*

- **Real-world:** the question's answer is not contained in the reviews but an educated guess can be made using common sense or real-world knowledge, e.g.

(7.4) **Question:** *Can I use the cloth to clean the keys on my clarinet?*

Answer Tag: Yes

(7.5) **Question:** *Has anyone traveled with this stroller on an airplane?*

Answer Tag: No

In (7.4) there is no reason to think the cloth could not be used for cleaning the keys of a clarinet as cloth can be used to clean most objects. Because strollers are usually not allowed inside an aircraft and collected by the airport employers at the gate, in (7.5) the answer *no* can be assumed.

- **Opinion:** the questions can be answered differently based on different reviews (7.6) or when the answer and review contain contradictory information (7.7). Often such questions ask for an opinion, so the answer depends on the user providing it, e.g.

  (7.6) **Review 1:** *...all in all these pans are worthless ...so many folks have had a horrid experience!!!*

  Review 2: *...At $15.00, it's a good pan for my purposes ...This pan is awesome for the price*

  Question: *is this item any good?*

  Answer Tag: *No*

  (7.7) **Review:** *...but it seems to get a little hot and makes a plastic noise under the sheet...*

  Question: *does it make noise when baby moves around?*

  Answer: *No not with a sheet on it.*

  Answer Tag: *No*

- **Unrelated:** the question is not about the product per se but about service and delivery, e.g (7.8). Such information is not product specific and is less likely to be mentioned in a review.

  (7.8) **Question:** *Is there a warranty when you buy it from amazon?*

- **No answer:** the questions which cannot be answered without additional information, i.e. the reviews do not contain the required information and common-sence reasoning is not sufficient to guess the answer (real-world).
- **Error:** the answer tag of the question contradicts the user-provided answer, e.g (7.9) where the answer is expressed in the negative form (*"I don't see why not..."*) but actually means *yes*, so the answer tag is wrong:

(7.9) **Question:** *Can you mount this upside down i.e. The receiver on top of the bumper?*
Answer: *I don't see why not, the is nothing preventing you.*
Answer Tag: *No* (Should be *Yes*)

I carried out this classification in a hierarchical fashion. For example, if the question cannot be answered straight away (the `straightforward` class) I first try to conclude the answer from provided information (`indirect`), and if that is not possible I try to guess the answer (`real-world`), and if that is also not possible, I mark it as a question which cannot be answered (`no answer`).

7.2.3 Availability of Answers

The *indirect* and *real-world* classes can be considered to be difficult questions. However, in general, I believe that the *straightforward*, *indirect* and *real-world* question classes can be answered without resorting to guessing outright. Other questions can still be answered randomly with a 50% probability of being correct. Formula (7.10) estimates the upper bound performance of a QA system:

$$Accuracy = \frac{|Q_{answerable}| + 0.5 * |Q_{guessing}|}{|Q|} \tag{7.10}$$

where $Q = Q_{answerable} \cup Q_{guessing}$ is a set of all questions, $Q_{answerable}$ is a set of answerable questions, and $Q_{guessing}$ is a set of questions which have to be guessed.

Detailed information is provided in Table 7.2. Approximately 8% of the data are errors. 53.5% of questions can be answered (36.5% are straightforward and 17% are difficult). Although it is difficult to conclude too much from this sample of 400, I can roughly estimate that the best performance I could expect from an automatic QA system would be around 77%. This means the system answers all answerable questions correctly (53.5%) and guesses half of those questions which cannot be answered (23.25%).

Domain	**Answerable**			**Guessing**				**Total (%)**
	STRF	**Indirect**	**RW**	**O**	**NA**	**U**	**Error**	
Home & Kitchen	31	9	4	9	34	1	12	100 (0.83)
Beauty	37	8	6	7	33	3	6	100 (3.6)
Baby	44	11	8	11	21	1	4	100 (4.6)
Clothing Shoes & Jewellery	34	8	14	10	23	1	10	100 (20.9)
Total	146	36	32	37	111	6	32	400

Table 7.2: Selection of 100 questions from 4 domains for manual analysis, where **STRF** – straightforward question, **RW** – real world question, **O** – opinion, **NA** stands for not answerable, and **U** – unrelated. The last column contains the percentage of the analysed questions from each domain (e.g. 100 is 4.6% of the Baby question data, 3.6% of Beauty, etc.).

7.3 Parallel work

The AmazonQA dataset is proposed by Gupta et al. (2019b) and based on the same data I use (He and McAuley, 2016; Wan and McAuley, 2016; McAuley and Yang, 2016). It contains 923k questions, 3.6M answers and 14M reviews about 156K products. The authors propose: 1) a method for selecting relevant reviews based on a combination of information retrieval techniques; 2) a model for generating an answer in a reading comprehension setting. Based on reviews, the authors mark each question as answerable or not, and conclude that more than half of all questions are answerable.

Altogether, the dataset contains 570,132 answerable questions and 15% of those are boolean. The average length of a question, answer and review is 14.8, 31.2, and 72.0 tokens respectively.

At the data processing step, Gupta et al. (2019b) remove duplicate text and length outliers (too long according to the median), tokenize text, divide it into snippets with the

length being either 100 tokens or the full sentence length, whichever is greater, then rank it based on TF-IDF using the BM25 function (Robertson and Jones, 1976; Robertson and Zaragoza, 2009). Only the top 10 snippets are kept and the rest of them are discarded. Then all the text is transformed to lower case except all capitalized words (for example, the acronym *"IBM"*) and each punctuation mark is presented as a single token, apart from apostrophes " ' ".

Gupta et al. (2019b) establish a MRC baseline with an LSTM-based encoder for reviews and questions. The review representation is aggregated and concatenated with the question representation. Then this concatenation is passed to the LSTM-based decoder to generate the token of the answer at each step. Additionally the R-Net proposed by Wang et al. (2017b)[8] was used to find the answer span in the reviews.

7.4 Data: AmazonYesNo version 2

Continuing work on the Amazon data in this section I build on my previous work (Dzendzik et al., 2019), the parallel work of Gupta et al. (2019b), and data[9] from Ni et al. (2019) as a continuation of the works of He and McAuley (2016); Wan and McAuley (2016); McAuley and Yang (2016). I made a number of changes to the data, compared to the version described in Section 7.2. The biggest disadvantage of the previous version of the dataset experiments was that it was not known whether a review contains an answer to the question. As Gupta et al. (2019b) proposed the classifier which filters unanswerable questions, I use their data[10] and work only with answerable Yes/No questions. I observe some questions in the dataset which were marked as answerable but the answer tag was not provided. I do not consider such questions. Also questions which contain both answers *"Yes"* and *"No"* are filtered out, as well as those answers which marked as *"somewhat"*. The statistics of the updated dataset are presented in Table 7.3.

[8] `https://www.microsoft.com/en-us/research/publication/mcr/` – last verified May 2021

[9] `http://deepyeti.ucsd.edu/jianmo/amazon/index.html` – last verified May 2021

[10] `https://github.com/amazonqa/amazonqa` – last verified May 2021

	All data			Balanced data		
Set	**# Q**	**# of Yes (%)**	**# Prod**	**# Q**	**# of Yes (%)**	**# Prod**
Train	44084	29880 (68%)	34336	28406	14202 (50%)	24075
Val	5692	3812 (67%)	4443	3760	1880 (50%)	3204
Test	5716	3844 (67%)	4396	3744	1872 (50%)	3115
Total	55492	37536 (68%)	43175	35190	17154 (50%)	30394

Table 7.3: Statistics for the AmazonYesNo dataset (v.2): Number of questions, products, and percentage of "Yes" answer

7.4.1 Passage Options

I consider a number of passage options to establish which one would works the best and what kind of patterns can be learned from the data.

- **Question only (No passage)**

  To see to what extent reviews are useful for answering user queries, I establish a baseline which tries to answer the question based only on the text of the question.

- **User-answer as a Passage**

  I use the answer-only setting to test the ability of the model to comprehend the user generated text. In this setting I can be confident the passage (answer provided by other users to the question) is definitely related to the question. In a way, in this setting I solve a very similar task to the Wan and McAuley (2016) by trying to classify answers as the answer "Yes" or the answer "No".

- **All Reviews (Theoretical)**

  Reviews contain a significant amount of information about the products. It seems reasonable to assume that user-generated reviews of the product can be used for answering user-generated questions. Theoretically, using the entire review would be the most complete scenario but due to the long review lengths and limited computational resources, I move to sentence extractions options instead.

- **Sentence Extraction Options**

  As can be seen from the data analysis (see Table A.2) the length of a review is significantly bigger compared to other RC datasets which makes the analysis of the

whole review more expensive. The Sentence Extraction option aims to reduce the number of sentences which come as input to a model.

- **AmazonQA Data**

  Finally, I use the snippets selected from reviews and provided by Gupta et al. (2019b). (The snippets were described in Section 7.3.)

The sentence extraction options need to be further clarified and I do so in the following section.

7.4.2 Sentence Extraction Options

There are multiple ways of extracting sentences relevant to the question from the passage. I focus on the following strategies: extracting *individual* sentences and selecting *continuous* sentences (window slide).

7.4.2.1 Individual Sentence Extraction

I split reviews into individual sentences[11] and concatenate together a list of sentences from all available reviews. Then I transform the question and all sentences to lower case, and remove punctuation and other non-character based symbols. I then calculate a sentence embedding and select the top m sentences most relevant to the question based on Euclidean distance and cosine similarity using the sentence transformer model[12] proposed by Reimers and Gurevych (2019) (as was described in Section 3.4.4.6). I select the top m sentences according to each distance and concatenate them together, deleting duplicates. This way I obtain a $k : 1 < k <= 2m$, unique individual independent sentences from reviews which are ranked as the most relevant to a question.

Additionally, I tried exactly the same sentence extraction mechanism with the small modification of removing stop words from the question and sentences before calculating

[11] I use the stanza library `https://stanfordnlp.github.io/stanza/` – last verified May 2021

[12] `https://github.com/UKPLab/sentence-transformers` – last verified May 2021

the embedding. Note, this has been done only to detect the most relevant sentences to the question but the set of original sentences is provided as a passage option.

7.4.2.2 Window Slide Extraction

The obvious disadvantage of selecting individual sentences as described above is the fact that any information which is spread across multiple sentences would be lost. To address this issue and still limit the length of input text I also try a *"window slide"* approach.

First I cover a long passage by windows with a fixed size of tokens. There are two parameters: (1) N – the number of tokens in the window, or in other words the length of the window; and (2) n – the offset, the number of the tokens I skip from the beginning of the previous window to the start of a new window. To ensure that we cover the full text and that a sentence which might contain the answer is not split by the window border, n should be much less than $N : n << N$. See Figure 7.3 for more details.

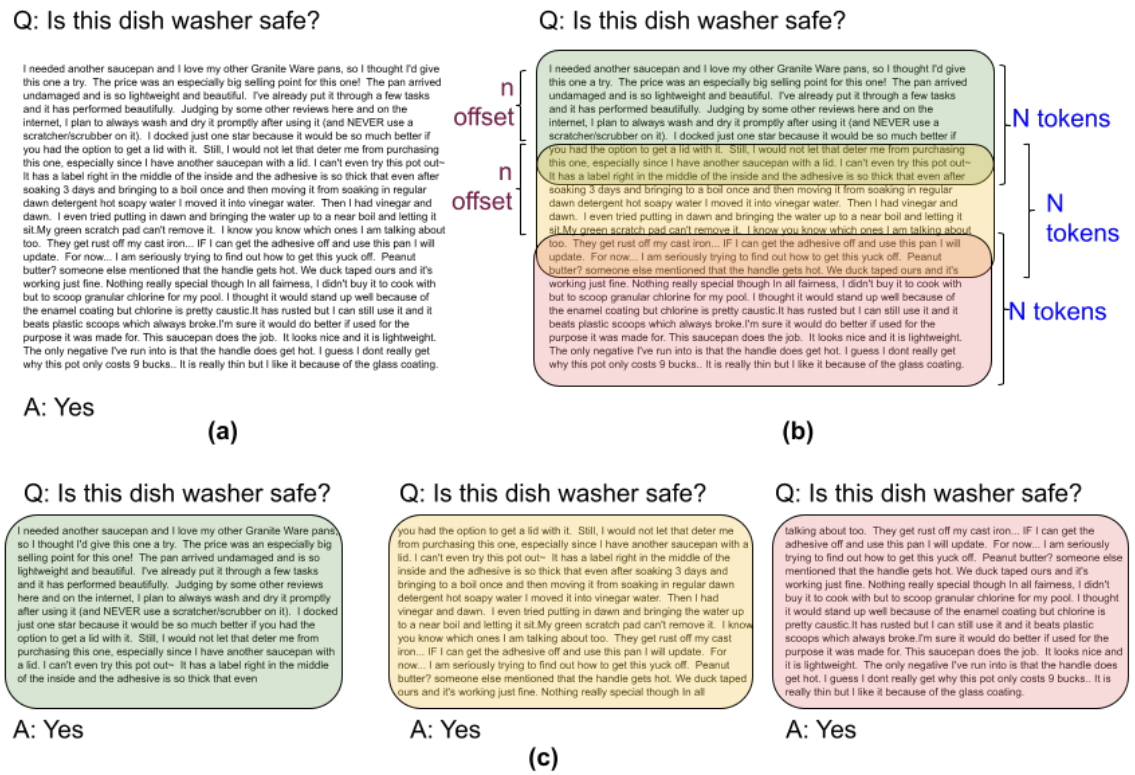


Figure 7.3: Process of selection of passages with the window slide approach. (a) illustrates the original review with the long text, (b) shows the usage of N-tokens and offset, and (c) represent individual windows for the question

Then the same process of selecting the most relevant set of windows is applied as

described before for the individual sentence selection. This way a number of continuous sentences from a review are obtained, which are also ranked as the most relevant part of a review to a question.[13]

7.5 Experiments

There are two main tasks: question answering based on the question only (*binary single-sentence classification task*) and reading comprehension, where there is a question and some passage (*entailment classification task* or pair sentence classification task). Those two tasks have been explored before and particularly discussed in Wang et al. (2018). I take advantage of the existing settings and use it in my approach. See Table 7.4 for more details .

Task name	**Classification Task**	**Passage adaptation**	**GLUE task reference**
Question only	single-sentence	None	CoLA (Warstadt et al., 2019)
Question + passage	sentence pair	Reviews (different forms) or answer	MNLI (Williams et al., 2018)

Table 7.4: Tasks description where **CoLA** is The Corpus of Linguistic Acceptability, **MNLI** is Multi-Genre Natural Language Inference. The word 'sentence' is used for convenience. Practically it can be more than one sentence.

My system architecture and code are mostly based on the HuggingFace implementation of $BERT$ transformer[14] which is based on PyTorch[15] as it gives more control over the random seed compare to TensorFlow. I focus my experiments on the `base` models and do not use the `large` models due to limited computational resources.[16]

[13] At earlier stages of this work I considered the possibility of using all question-window pairs as individual data samples. In other words, each window would produce a new unique question-passage sample which could be fed into the model individually. This way, out of one data sample, I get several data samples for one question. That means several input instances for the model. Unfortunately, this approach creates a lot of noisy data where the passage has no question-related information. This is especially critical during the training phase, so I rejected this idea.

[14] `https://github.com/huggingface/transformers` – last verified May 2021

[15] `https://pytorch.org/` – last verified May 2021

[16] The large model requires significantly more memory compared to the base model. To fit the large model into the memory a significant reduction of sequence length and/or batch size would be needed. Turc et al. (2019) studied 24 pre-trained miniature BERT models and also comment on this issue pointing out that for running the large model for SQuAD with the length of 256 the maximum batch size is 2, which is so small

I run all experiments on balanced data with 10 random seeds and evaluate *accuracy*, *stable accuracy* (introduced in Chapter 5), *strict majority voting* (a answer is counted as correct only if the strict majority of models selected the correct answer), *random majority voting* (random answer selection if there is an equal amount of votes for an answer), and the *confidence ensemble* which selects the answer based on the output probability (the answer which has the highest probability across runs is selected). Accuracy is reported as the average across 10 runs, while all other metrics are reported as accumulated, i.e. they are calculated with consideration of predictions from 10 models, if not specified otherwise.[17]

7.5.1 Answer Only

In the first set of experiments, I test the performance of an out-of-the-box $BERT$ model in the answer only setting. In this scenario, I use real answers provided by users to the question. It is done to (1) verify that the approach is applicable in general, and (2) verify the gold answer tags. The average results over 10 runs are presented in Table 7.5. All five evaluation metrics show very high results $> 99.96\%$, which indicates that the model is capable of answering spontaneously user generated data with perfectly suitable passages (the answer in this case). I looked into those samples which were answered incorrectly and will discuss it in the Error Analysis section (Section 7.6).

	Acc	**Stable Acc**	**MV**	**Random MV**	**Confidence**
Answers	99.96	99.87	99.97	99.97	99.96

Table 7.5: Results over 10 runs for answer setting. Where **Acc** stands for accuracy, and **MV** stands for majority voting.

that it will actually harm the performance. Reducing the passage length further than 300 tokens make the comprehension part of the task extremely limited and unsatisfying in terms of my research interests. More details here: `https://github.com/google-research/bert#out-of-memory-issues` – last verified May 2021

[17]The results for individual runs in every reported experiment are omitted in this work but available by request.

7.5.2 Question Only

To set up the baseline I run the question-only setting. The average results over 10 runs are presented in Table 7.6. Noticeably, the performance of all metrics except the stable accuracy is significantly better than the random baseline of 50% (single run) as data is balanced.[18] I can interpret this to mean that the question alone contains some information what can be used to provide an answer.

	Acc	**Stable Acc**	**MV**	**Random MV**	**Confidence**
Question only	61.38	35.43	60.24	62.34	61.89

Table 7.6: Results over 10 runs for question only setting.

7.5.3 Passage Options

I experiment with the following passage options:

- the top rated snippet from AmazonQA (*top 1 snippet*);
- top 3 snippets from AmazonQA (*top 3 snippets*);
- top 10 snippets from AmazonQA (*all 10 snippets*);
- the most relevant window extracted from the original reviews (size $N = 100$ and offset $n = 50$) (*window*);
- the most relevant window ignoring the stop words (*window - SW*);[19]
- the most relevant sentences extracted $m = 1$ (*1 sent*);
- the top related sentences without stop words (*1 sent - SW*);

[18]Clearly, the stable accuracy of 10 runs is 0.5^{10} which is smaller than stable accuracy of presented results. It does not make sense to me to run random baseline more than once as there is no stability, that is why I compare to the single run.

[19]The stop words are removed only for calculating the similarity with question and selecting the most relevant option. Than the selected sentence (or number of sentences) are used as the paragraph option in the original way with the stop words in it.

- the top 3 sentences ($m = 3$) without stop words (*3 sent - SW*);[20]
- a combined option of window slide and sentence extraction (*sent + window*).

	Train					Validation				
	sent	sent -SW	window	window -SW	Any	sent	sent -SW	window	window -SW	Any
Total	28406					3760				
Exact Inclusion										
Top 1	0.10	0.11	0.05	0.06	0.20	0.10	0.12	0.06	0.06	0.22
Top 3	0.21	0.23	0.10	0.11	0.38	0.22	0.24	0.11	0.12	0.40
All 10	0.38	0.42	0.18	0.22	0.62	0.42	0.45	0.21	0.24	0.66
Bag-of-Words ratio										
AVG	0.73	0.76	0.7	0.73	-	0.76	0.78	0.73	0.74	-
$> 80\%$	0.45	0.50	0.37	0.40	-	0.50	0.53	0.41	0.43	
$< 20\%$	0.03	0.04	0.03	0.03	-	0.02	0.04	0.03	0.03	

Table 7.7: Top: Proportion of extracted sentences which are exactly included into AmazonQA snippets (top 1, top3 and all 10). Bottom: the average BOW ratio, proportion of sentences were BOW overlap is over 80% and lower than 20%. Where **Any** is counts if any of considered options is presented in the AmazonQA snippets.

Table 7.7 contains detailed information on the overlap between extracted sentences and the AmazonQA snippets. This estimation shows that sentences extracted individually or via "windows slide" approach are different enough to the snippets and it worth investigating the performance on this data. The inclusion of exact sentence extracted (by any method) from reviews into all 10 AmazonQA snippets for train and validation sets occurs only in 62%-66%. Bag-of-Words shows the average overlap to be between 70%-78% of tokens. I also looked into the number of cases where the overlap is high (>80%) and relatively low (<20%) compared to all 10 snippets. Up to half of the passages have over 80% BOW overlap with the snippets from AmazonQA. Additionally, only 4% of passages have an overlap of under 20%.

The results are presented in Table 7.8. For the convenience of comparison, I repeat the *question only* baseline result at the top of the table.

The top three snippets from AmazonQA show the best results across all metrics. The

[20]Note, I select one (three) sentences according to two similarities, if those sentences are different the concatenation might contain up to six sentences.

	Accuracy	**Stable Accuracy**	**MV**	**Random MV**	**Confidence**
Question only	61.38	35.43	60.24	62.34	61.89
AmazonQA Data					
- top 1 snippet	63.00	34.49	61.68	64.00	64.63
- top 3 snippets	**64.37**	**35.48**	**63.59**	**65.89**	**65.35**
- all 10 snippets	64.28	35.61	63.46	65.55	65.82
Review Extraction					
- window	61.43	31.94	60.05	62.54	61.94
- window - SW	60.96	31.86	59.95	62.10	61.89
- 1 sent	61.12	30.72	59.76	62.29	61.49
- 1 sent - SW	61.74	33.24	60.77	62.97	63.01
- 3 sent	61.40	33.44	60.45	62.10	62.49
- 3 sent - SW	62.77	33.96	61.99	63.98	63.32
- sent + window	62.48	34.68	63.59	63.59	63.62

Table 7.8: The result of averaged accuracy of out-of-the-box transformer (bert-based-uncased) solution for AmazonYesNo. **MV** stands for Majority Voting, *-SW* stands for removing stop words from the sentence on the sentence selection stage.

second best is a concatenation of all AmazonQA snippets.[21] The window slide approach (with and without stop words), single sentence selection, and top 3 sentence selection do not outperform the question only-baseline. Removing the stop words does not improve results for window slide but increases accuracy for the top one selected sentence and top 3 selected sentences. Also, a combination of extracted window and sentences provides a better stable accuracy than those paragraphs individually but does not outperform them in average accuracy, majority voting, and confidence ensemble.

7.5.4 Alternative Architectures and Transfer Learning

Selecting those paragraph options which show the best result in each qualitatively different setting (different passages but not modifications such as changing the number of sentences or use of the stop words), I tried an alternative to the BERT model – RoBERTa. The results are shown in Table 7.9 The number of models is indicated in the brackets. Using the same random seeds from the previous experiments I observe that in some cases the model is not able to learn anything. Those runs were excluded from the average ac-

[21] Those two settings are very close as the model takes only the first 300 word-pieces which are close to the 3 snippet length.

	Acc	Stable Acc	MV	Random MV	Confidence
	RoBERTa base				
Question only (8)	59.93	42.93	57.95	60.01	60.56
top 3 snippets (2)	57.70	37.71	37.71	57.70	60.88
window (3)	53.87	28.64	54.81	54.81	54.89
3 sent - SW (4)	57.11	33.24	49.76	57.78	59.55
sent + window (4)	56.69	34.41	50.93	57.43	57.53

Table 7.9: The result of averaged accuracy of out-of-the-box transformer solution for AmazonYesNo with RoBERTa base model.

curact evaluation which leaves a smaller number of models to average. The results of the RoBERTa models are lower for all metrics than the BERT models except the stable accuracy but that is due to the fewer number of models.

I also tested one BERT model and two RoBERTa models which were fine-tuned for sequence classification on additional data: IMDB dataset,[22] and MNLI.[23] All three of those models are provided by Morris et al. (2020). The MNLI dataset is not a binary classification problem but has three options including neutral. I ignore the neutral class and apply the model to the question-only baseline and passage options from previous experiments, apart from the window slide approach as it shows the lowest performance.

The results are presented in Table 7.10. The RoBERTa based MNLI model shows the best performance for all five metrics for both question-only setting (achieving 62,76% with random majority voting), and top 3 snippets achieving (66.67% with confidence ensemble).

7.5.5 Ensemble

In the previous subsection I ensemble models over random seeds keeping different models and data (different passage options) separately. In this subsection I ensemble results of over 120 models across architectures and data.

Table 7.11 presents the averaged results across models and data sources. For all settings, the random majority voting shows the best results compared to the confidence en-

[22] `https://huggingface.co/textattack/roberta-base-imdb` – last verified May 2021
[23] `https://huggingface.co/textattack/roberta-base-MNLI`– last verified May 2021

	Acc	Stable Acc	MV	Random MV	Confidence
RoBERTa base + IMDB					
Question only	61.07	36.25	60.72	62.57	62.70
top 3 snippets (9)	62.80	21.54	66.20	66.20	66.38
3 sent - SW (9)	60.87	18.75	64.47	64.47	65.08
sent + window (10)	62.38	28.54	61.62	63.94	63.97
BERT base + MNLI					
Question only	60.78	33.64	59.65	62.18	61.65
top 3 snippets	64.69	35.40	64.41	66.48	65.61
3 sent - SW	61.36	32.42	60.13	62.38	62.26
sent + window	62.90	34.02	61.52	63.86	63.64
RoBERTa base + MNLI					
Question only	62.14	39.89	61.52	63.10	62.34
top 3 snippets	**65.51**	**39.02**	**64.71**	**66.54**	**66.68**
3 sent - SW	62.57	34.41	61.91	63.96	63.83
sent + window	64.41	37.18	64.20	66.13	65.03

Table 7.10: The result of averaged accuracy of out-of-the-box transformer solution (BERT and RoBERTa) for AmazonYesNo and Transfer Learning (IMDB and MNLI).

semble. The question only setting achieve average performance of 61.11% and 63.41% with random majority voting. Across different models with the different passage options, top 3 snippets show the best performance of 64.06% average accuracy and 66.8% average accuracy with random majority voting. The average accuracy result with top 3 snippets from AmazonQA is statistically significantly better than question-only.[24] The random majority voting accuracy difference is also statistically significant.[25]

	Acc	Stable Acc	MV	Random MV	Confidence
Question only:	61.11	12.01	63.41	63.41	62.50
Top 3 AmazonQA snippets	**64.06**	10.64	**66.80**	**66.80**	66.92
3 sent - SW	61.47	15.31	63.85	63.94	62.85
Joined sent+window	62.47	13.06	64.59	65.45	64.87
Combined models and Passage Options	62.64	0.43	**67.73**	**67.73**	66.74

Table 7.11: The result of averaged performance of ensemble models for AmazonYesNo.

Combining together all models and passage options I observed the average accuracy is only 62.64% but it is still significantly better than the question only setting.[26]. The highest

[24]According to T-test for two independent means: $t - value = -6.47346$, $p - value < .00001$.
[25]According to Z-score: $z - score = 3.158$, index control is 95%.
[26]According to T-test for two independent means: $t - value = -3.73586$, $p - value = 0.000254$

average accuracy is 67.73% and obtained with random majority voting. At the same time average accuracy for all architecture/passage combinations is significantly lower than using only AmazonQA data (62.64% vs. 64.06%)[27]. The improvement of accuracy in random majority voting of combined passages over the top 3 AmazonQA snippets is not statistically significant.[28]

Note that the stable accuracy over all runs is only 0.43%. Less than one percent of questions are answered correctly by all 127 models. While the individual models have a different sets of questions they answer correctly, when ensembles together the models do not find an agreement and at least one model provide an incorrect answer for a significant number of questions. Instead of achieving a high performance together, the models "agree to disagree" with each other.

7.5.6 Performance on Test Set

Although the best performance was achieved over a combination of models and passage options, it required fine-tuning 100+ models.[29] I evaluate the RoBERTa model fine-tuned on MNLI on the test set using AmazonQA top 3 snippets as it showed the best average accuracy and the improvement of the combined approach with random majority voting is not statistically significant on the development set. The same as before, the averaged results over 10 runs are presented in Table 7.12.

	Acc	**Stable Acc**	**MV**	**Random MV**	**Confidence**
Question only	61.34	39.34	60.36	62.01	62.37
Top 3 AmazonQA snippets	65.56	38.06	64.96	**67.04**	65.89

Table 7.12: The result of averaged performance of RoBERTA+MNLI model for AmazonYesNo on test set.

All three, average accuracy, random majority voting, and confidence ensemble are statistically significantly better than the question-only setting.[30] And again, the best average

[27] According to T-test for two independent means: $t - value = 2.81323$, $p - value = 0.005497$

[28] According to Z-score: $z - score = 0.452$, index control is 95%.

[29] It takes 4-8 hours to tune one model on the GPU NVIDIA RTX 2080ti with 256GB RAM.

[30] Average: $t - value = -18.10951$; $p - value < 0.00001$, random majority voting: $Z - score = 4.141$, and confidence ensemble: $Z - score = 3.998$.

performance 67.04% is achieved by random majority voting.

7.6 Error Analysis

In this section I will present the results of an error analysis which was manually carried out on the development set.

7.6.1 Answer-Only

I start with the answer-only setting, where system predicts *yes* or *no* based on the text of user provided answer. There is one single question all the BERT models get wrong - see Example (7.11). The answer explicitly says yes but the rest of the text contains sarcasm, so it might indeed look confusing.

(7.11) **Question:** *If my cat brushes against it, will it turn on?*

Passage (User Answer): *yes....but only if you don't want it to...I would not buy this again.*

Answer Tag: No

Predicted Answer: Yes

A further nine examples are marked as mixed, which means that some models answer this question correctly while some provided the wrong answer. Only one of those questions is provided with a clear answer and the answer tag is wrong. Another 5 questions do not have a confident answer but use phrases such as:

- *I'd say, No, it won't work for you.*;
- *Quite frankly I don't see why you could not use this*;
- *I'm not positive it will work...*;
- *... if this is the case then yes...*;
- *I do believe this one does as well..*

For all those samples, at least one model predicted the wrong answer tag. One question actually contains two questions in it but only one is answered by the user. The final two questions contain a lot of on-topic discussion where the direct answer is not provided but has to be understood from the context. All 9 samples, regardless whether the direct answer is provided or not, contain a description of some experience with the product.

7.6.2 Question-Only

I next analyse the setting where only the question is provided. I focus on the combination of model fine-tuned on MNLI as it shows the best performance.

When observing stable correct and stable errors samples, I have noticed a vocabulary bias. The model tends to provide the answer "yes" to those questions where the words *works* and *use* are mentioned. At the same time, it tends to provide the answer "no" to those questions where the question contains the words *come with* – see Figure 7.4.[31]

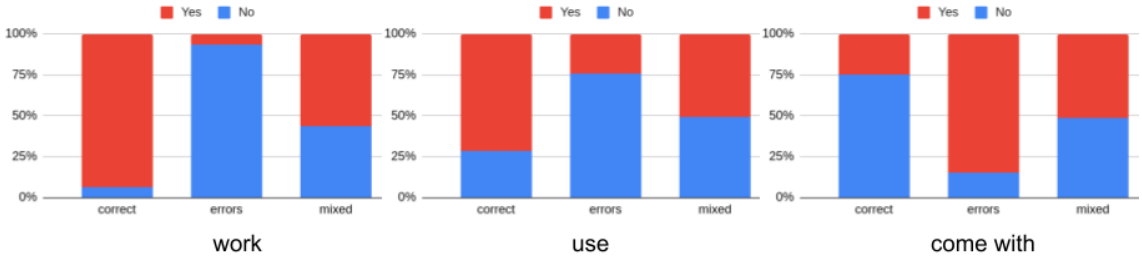


Figure 7.4: Distribution of the answers "Yes" and "No" from questions which containing words *work, come with*, and *use* across stable correct (errors), and mixed samples.

7.6.3 Amazon Top 3 Snippets

I examine the output of the AmazonQA top-3 snippets+MNLI system as it showed the best performance.

I inspected 20 samples from stable errors and found that 2 samples had a wrong answer tag. 10 passages did not contain the answer to the question, and another 5 samples

[31]The bias towards the word *work* was first observed accidentally by examining a few stable errors samples and then was verified automatically across all samples. Motivated by this I discovered two other words which bias the answer. There could be other words and dependencies but it is not trivial to identify this so I draw the line here. Table B.1 in the Appendix B presents a number of questions which contains the mentioned words across samples which were answered stably correct, stably wrong, and mixed.

provided relevant content but did not provide a specific answer. Questions like that are common enough and asking about a compatibility with some other product or equipment like *would this case fit my *particular model of the phone** or *would this battery work with my *particular model of laptop**, etc. It is difficult to answer those questions unless it was explicitly mentioned in a review, or without additional and very specific knowledge about the product. In two other samples, the answer in the passage is expressed using negation, e.g. (7.12). It looks like in this case the model ignored negation. The final sample is confusing and the question is difficult for comprehension and answering based on the review.

(7.12) **Question:** *My laptop model is Toshiba Satellite L775, is this show battery is right. I need to replace the battery right now.*

Passage: *... The battery doesn't even fit the compartment of my Toshiba Satellite and I was very specific in which model I had. ...*

Answer: No

7.7 Summary

In this chapter, firstly, I gave an overview of datasets built based on user-generated queries and reviews. Secondly, I introduced a fully user-generated reading comprehension dataset by combining two existing datasets into a new one designed to address yes/no questions about products using reviews. I provided an analysis of data and showed that reviews can, to some extent, be used to answer yes/no questions. I investigated a number of passages options, and two transformer-based architectures, combined with different pre-training tasks for answering user generated questions. Also, I carry out an error analysis of the best performing system.

The main findings are:

1. Consumer reviews can be used to *some extent* to answer questions about products as not all questions are answered in reviews. It is still a challenging task to find and understand the answer if it is in the review;

2. The question only setting performs better then the random baseline which indicates that the question itself has some information about answers;

3. The setting which use the top 3 snippets from the AmazonQA dataset shows the best result where 3 snippets align with the maximum length, I can feed into the model;

4. Using a previously fine-tuned model on MNLI data improves the result for both the BERT and RoBERTa model;

5. A voting ensemble approach over all passages options and model architectures gives the highest performance;

6. Ensemble by confidence does not improve over ensemble by voting;

7. A manual error analysis reveals that some errors are related to the fact the AmazonQA snippets do not contain the information which is needed to provide the answer to the question.

A description of the first version of the AmazonYesNo dataset and some preliminary experiments has been published at The Student Research Workshop 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Dzendzik et al. (2019)).[32] The dataset is publicly available by request.

[32] `https://aclanthology.org/N19-3001/` – last verified November 2021

Chapter 8

Conclusion and Future Work

> But you can't read every story, and
> answer every question even if
> you'd like to
>
> *Lemony Snicket*
>
> *The End*

In this thesis I provided extensive analysis of MRC datasets and then explored the English Reading Comprehension Question Answering task with a focus on multiple-choice and boolean questions. I primarily addressed the tasks using string similarities, transformer, and graph-based architectures. My experiments showed that those approaches can provide reasonable performance on these tasks.

Looking at the MRC task from a distance I can conclude the following: (1) the string similarity approach works well for texts with significant lexical overlap between question/answer and passage but does not handle paraphrasing well; (2) pretrained transformer models show good performance on MRC tasks with clean data but tend to be unstable; (3) spontaneously user-generated data requires a more specific approach due to the high amount of typos, grammatical errors, specific vocabulary, and slang used in the provided text.

In this final chapter, I first revisit the research questions which were posed in Chapter 1, and then indicate several directions for future work.

8.1 Research Questions Revisited

I summarise the contribution of my thesis according to the research questions.

RQ 1: What kind of reading comprehension datasets are available, what kind of tasks do they cover, and what makes a dataset difficult?

In Chapter 2 I provide a full classification of question and answer types for the MRC task and describe how they are related to each other. I present a systematic overview of over 50 English reading comprehension datasets, including basic statistics, vocabulary analysis, and named entity analysis. I conclude there are a number of data sources that are used for MRC such as Wikipedia, books, news, blogs, as well as automatically generated, crowdsourced, and spontaneously user-generated sources. I show how the datasets overlap in data, and particular dataset samples. I also discovered that the questions *"Why?"*, *"When?"* and *"Where?"* are particularly underrepresented in existing datasets. I addressed the question *"What makes a dataset difficult?"* by applying a regression model to predict human performance and state-of-the-art results. I established that average passage length, automatic generation of content, and multi-hop questions have a correlation (either positive or negative) with both SOTA and human performance. Finally, I show the difference between MRC datasets created for people and those which were created for automatic reading comprehension.

RQ 2: How can a combination of features based on the similarity of natural language representations impact the state-of-the-art results in a multiple-choice reading comprehension task?

In Chapter 4, I show how different types of string similarities described in Chapter 3 can be combined together to achieve (at the time) the state-of-the-art performance in an answer selection task. The proposed method achieved the best performance at the time for the MovieQA dataset (Tapaswi et al., 2016) (the plot setting) and the MCQA Shared Task (Shangmin et al., 2017).

Moving from multiple-choice to the boolean questions, the third research question is about these types of questions and particularly about the use of knowledge graphs in

MRC.

RQ 3: How can knowledge graph information be used to impact the state-of-the-art methods in the boolean reading comprehension task?

In Chapter 5 I provide a motivation for answering boolean questions. I look into the BoolQ dataset (Clark et al., 2019) and carry out an error analysis, discovering the unstable behavior of the BERT baseline (Devlin et al., 2019).

In Chapter 6 I explore possible combinations of additional knowledge resources, including ConceptNet (Speer et al., 2017) and Google KG, in an ultimately unsuccessful attempt to improve performance in boolean MRC. The possible reasons for the lack of improvement are:

- ConceptNet often contains too general information and has low entity coverage;
- a GNN, although more suitable for modeling graph-based input, does not improve over our baseline model probably due to the inaccuracies in the entity linking and KG coverage.

Finally, in the last research question I move from crowdsourcing data towards spontaneously user-generated data.

RQ 4: How do state-of-the-art MRC approaches perform with user generated data in the boolean reading comprehension task?

In Chapter 7 I presented a new fully user generated dataset based on the Amazon dataset provided by He and McAuley (2016); Wan and McAuley (2016); McAuley and Yang (2016). I analyse the questions in this dataset with a focus on their answerability in the corresponding user reviews. First, working on original data, and then, building on the parallel work of Gupta et al. (2019b), I established a transformer-based baseline for naturally occurring texts and questions. I also compared a number of passage options and determine that the top 3 snippets provided by Gupta et al. (2019b) helps the most to answer the questions.

8.2 Possible Future Work

In this section I describe how the work in this thesis might be extended and discuss what kind of future research can be done to contribute to the field of English Machine Comprehension Question Answering.

8.2.1 Reading Comprehension Datasets Overview

I presented a broad overview of English Reading Comprehension Question Answering datasets in Chapter 2, limiting the survey to those which are appeared before the middle of the year 2020. As reading comprehension and question answering remain the focus of interest, almost every month a new dataset is announced which can be incorporated into the survey with the corresponding statistics and analysis. I also limited the analysis to data source, creation methods, question words, vocabulary, and named entities. There are deeper analyses which could be done with a closer look into the grammatical structure of questions and passages, and possible biases in data.

Dunietz et al. (2020) raise the important issue of reading comprehension dataset complexity. Although I looked into dataset complexity from the point of human performance and SOTA, more analysis could be done here. Additionally, more datasets and characteristics could be included in the comparison of datasets created for people with those which were created for MRC. Deeper understanding of this difference might make future MRC datasets more challenging and improve natural language understanding.

It might be worthwhile to combine my survey with the work of Wang (2020); Zeng et al. (2020); Baradaran et al. (2020).

A similar type of analysis can be done for other reading comprehension and/or question answering tasks, including datasets in other languages.

8.2.2 Similarity and Linguistic Features

In Chapter 4 I explored approaches for multiple choice question answering based on string similarities. There is room there for the inclusion of linguistic analysis such as part-of-

speech tagging, discourse analysis, negation, and implicative word analysis. In many neural approaches, there is a certain reliability on the model to implicitly learn by such representations but more ways of combining existing features could be explored.

8.2.3 Stability

In Chapter 5 I discovered the instability of BERT-based models on the BoolQ dataset. A similar analysis could be done for other non-boolean MRC datasets and methods.

Inspired by Toneva et al. (2019), who explored "forgettable" examples in image recognition, I would like to indicate a research direction for "forgettable" and "learned" examples in reading comprehension. Toneva et al. (2019) highlights that forgettable examples seem to exhibit peculiar or uncommon features. A discovery of such features in reading comprehensions would be beneficial for natural language understanding.

8.2.4 Knowledge Graphs

In Chapter 6 I applied external knowledge graph information to the BoolQ dataset (Clark et al., 2019) and did not observe significant improvements. The idea of combining external knowledge with plain text is still reasonable and can be explored with more suitable larger sized data, better entity coverage, recognition, and linking. Also, alternative architectures for incorporating external knowledge can be explored.

8.2.5 Spontaneously Generated Data

As there is a huge amount of user data generated every day online, it is natural to expect there to be interest in understanding and processing such content. People express themselves in different ways making some amount of grammatical mistakes and typos, using slang and short abbreviations. This type of text differs from edited text such as books, news articles, and even Wikipedia articles. Suitable modifications to existing models and new architectures may need to be explored.

8.3 Final Remarks

I believe more effective solutions to the machine reading comprehension problem will require deeper research and understanding of what kind of answers we expect to be provided by reading comprehension systems.

AI-based systems have already become a part of people's life. I would like to highlight two related points which I found important to understand in order to build an effective reading comprehension system:

- ***Know your data.***

  It is absolutely necessary for any type of data-driven approach to know how big, accurate, representative, and complex the data is. It will help to select the appropriate approach and understand the task better.

- ***Know your errors*** or ***Is the answer correct for the correct reason?***

  Given a labelled test set, it is straightforward to determine whether a system has answered a questions correctly. However, it is more important to understand *why* the system gives this answer. To the best of my knowledge, there is no easy way do this, although there is currently an active area of research (Ribeiro et al., 2016; Belinkov et al., 2020; Jain and Wallace, 2019; Wiegreffe and Pinter, 2019).[1] In the production systems wrong reasoning can and will lead to the wrong answers. A detailed and systematic analysis of a system performance (error and correct) is a good place to start.

Beyond a shadow of a doubt, developing a MRC system that would be able to deal with all sorts of questions in different domains is a very challenging, interesting, and still unsolved task. It involves genuine natural language understanding and has many possible directions for further research.

[1] `https://interpret-neural.github.io/2021/01/28/tutorial.html` – last verified November 2021

Bibliography

Rami Al-Rfou, Marc Pickett, Javier Snaider, Yun hsuan Sung, Brian Strope, and Ray Kurzweil. 2016. Conversational contextual cues: The case of personalization and history for response ranking. *arXiv:1606.00372*.

James A. Anderson. 1972. A simple neural network generating an interactive memory. *Mathematical Biosciences*, 14(3):197–220.

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

Akari Asai, Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2018. Multilingual extractive reading comprehension by runtime machine translation. *arXiv:1809.03275*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 722–735. Springer.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ondrej Bajgar, Rudolf Kadlec, and Jan Kleindienst. 2017. Embracing data abundance. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*.

Razieh Baradaran, Razieh Ghiasi, and Hossein Amirkhani. 2020. A survey on machine reading comprehension systems. *arXiv:2001.01582*.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230, Brussels, Belgium. Association for Computational Linguistics.

Yonatan Belinkov, Sebastian Gehrmann, and Ellie Pavlick. 2020. Interpretability and analysis in neural NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 1–5, Online. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Johannes Bjerva, Nikita Bhutani, Behzad Golshan, Wang-Chiew Tan, and Isabelle Augenstein. 2020. SubjQA: A Dataset for Subjectivity and Review Comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5480–5494, Online. Association for Computational Linguistics.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3(null):993–1022.

Matthias Blohm, Glorianna Jagfeld, Ekta Sood, Xiang Yu, and Ngoc Thang Vu. 2018. Comparing attention-based convolutional and recurrent neural networks: Success and limitations in machine reading comprehension. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 108–118, Brussels, Belgium. Association for Computational Linguistics.

Dasha Bogdanova and Jennifer Foster. 2016. This is how we do it: Answer reranking for open-domain how questions with paragraph vectors and minimal feature engineering. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016, pages 1290–1295.

Dasha Bogdanova, Jennifer Foster, Daria Dzendzik, and Qun Liu. 2017. If you can't beat them join them: Handcrafted features complement neural nets for non-factoid answer reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 121–131, Valencia, Spain. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Francis Bond and Ryan Foster. 2013. Linking and extending an open multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia, Bulgaria. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv:1506.02075*.

Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 628–635, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Kevin Burton, Akshay Java, and Ian Soboroff. 2009. The ICWSM 2009 Spinn3r Dataset. In *Third Annual Conference on Weblogs and Social Media (ICWSM 2009)*, San Jose, CA. AAAI.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 1306–1313. AAAI Press.

Casimiro Pio Carrino, Marta R. Costa-jussà, and José A. R. Fonollosa. 2020. Automatic Spanish translation of SQuAD dataset for multi-lingual question answering. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5515–5523, Marseille, France. European Language Resources Association.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Atef Chaudhury, Makarand Tapaswi, Seung Wook Kim, and Sanja Fidler. 2019. The shmoop corpus: A dataset of stories with loosely aligned summaries. *arXiv:1912.13082*.

Danqi Chen. 2018. *Neural Reading Comprehension and Beyond*. Ph.D. thesis, Stanford University.

Michael Chen, Mike D'Arcy, Alisa Liu, Jared Fernandez, and Doug Downey. 2019. CODAH: An adversarially-authored question answering dataset for common sense. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 63–69, Minneapolis, USA. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-fine question answering for long documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 209–220, Vancouver, Canada. Association for Computational Linguistics.

Julie Medero Kazuaki Maeda Christopher Walker, Stephanie Strassel. 2006. Ace 2005 multilingual training corpus.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv:1803.05457*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA. Association for Computing Machinery.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45.

Nick Craswell. 2009. *Mean Reciprocal Rank*, pages 1703–1703. Springer US, Boston, MA.

Danilo Croce, Alexandra Zelenanska, and Roberto Basili. 2018. Neural learning for question answering in italian. In *AI*IA 2018 – Advances in Artificial Intelligence*, pages 389–402, Cham. Springer International Publishing.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 593–602. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Frederick B. Davis. 1944. Fundamental factors of comprehension in reading. *Psychometrika volume*, 9:185–197.

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

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017a. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1832–1846, Vancouver, Canada. Association for Computational Linguistics.

Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017b. Quasar: Datasets for question answering by search and reading. *arXiv:1707.03904*.

Martin d'Hoffschmidt, Wacim Belblidia, Quentin Heinrich, Tom Brendlé, and Maxime Vidal. 2020. FQuAD: French question answering dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1193–1208, Online. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dunietz, Greg Burnham, Akash Bharadwaj, Owen Rambow, Jennifer Chu-Carroll, and Dave Ferrucci. 2020. To test machine comprehension, start by defining comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7839–7859, Online. Association for Computational Linguistics.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A new Q&A dataset augmented with context from a search engine. *arXiv:1704.05179*.

Daria Dzendzik, Jennifer Foster, and Carl Vogel. 2021. English machine reading comprehension datasets: A survey. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8784–8804, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daria Dzendzik, Alberto Poncelas, Carl Vogel, and Qun Liu. 2017a. ADAPT Centre Cone Team at IJCNLP-2017 Task 5: A similarity-based logistic regression approach to multi-choice question answering in an examinations shared task. In *Proceedings of the IJCNLP 2017, Shared Tasks*, page 67–72. Asian Federation of Natural Language Processing.

Daria Dzendzik, Carl Vogel, and Jennifer Foster. 2019. Is it dish washer safe? Automatically answering "Yes/No" questions using customer reviews. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 1–6, Minneapolis, Minnesota. Association for Computational Linguistics.

Daria Dzendzik, Carl Vogel, and Jennifer Foster. 2020. Q. Can knowledge graphs be used to answer Boolean questions? A. It's complicated! In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 6–14, Online. Association for Computational Linguistics.

Daria Dzendzik, Carl Vogel, and Qun Liu. 2017b. Who framed Roger Rabbit? Answering questions about movie plot. The Joint Video and Language Understanding Workshop: MovieQA and The Large Scale Movie Description Challenge (LSMDC), at ICCV 2017, 23th of October, Venice, Italy.

Pavel Efimov, Andrey Chertok, Leonid Boytsov, and Pavel Braslavski. 2020. SberQuAD – Russian Reading Comprehension Dataset: Description and Analysis. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 3–15, Cham. Springer International Publishing.

Lisa Ehrlinger and Wolfram Wöß. 2016. Towards a definition of knowledge graphs. In *SEMANTiCS (Posters, Demos, SuCCESS)*, pages 13–16.

Ahmed Elgohary, Chen Zhao, and Jordan Boyd-Graber. 2018. A dataset and baselines for sequential open-domain question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1077–1083, Brussels, Belgium. Association for Computational Linguistics.

Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. 2017. Linked Data Quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web Journal*, pages 1–53.

Christiane Fellbaum and George A. Miller. 1998. *WordNet: An Electronic Lexical Database*. Language, speech, and communication. MIT Press.

Alena Fenogenova, Vladislav Mikhailov, and Denis Shevelev. 2020. Read and reason with MuSeRC and RuCoS: Datasets for machine reading comprehension for Russian. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6481–6497, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yaroslav Fyodorov, Yoad Winter, and Nissim Francez. 2000. A natural logic inference system. In *Proceedings of the 2nd Workshop on Inference in Computational Semantics (ICoS-2)*.

Yaroslav Fyodorov, Yoad Winter, and Nissim Francez. 2003. Order-Based Inference in Natural Logic. *Logic Journal of the IGPL*, 11(4):385–416.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural message passing for quantum chemistry. volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272, International Convention Centre, Sydney, Australia. PMLR.

Taisia Glushkova, Alexey Machnev, Alena Fenogenova, Tatiana Shavrina, Ekaterina Artemova, and Dmitry I. Ignatov. 2020. DaNetQA: a yes/no question answering dataset for the russian language. In *Proceedings of the 9 th International Conference on Analysis of Images, Social Networks and Texts, AIST 2020*, pages 57–67. Springer.

Aaron Gokaslan and Vanya Cohen. 2019. OpenWebText Corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Wael H. Gomaa and Aly A. Fahmy. 2013. Article: A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv:1410.5401*.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2019a. A deep neural network framework for English Hindi question answering. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 19(2):Article 25.

Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary C. Lipton. 2019b. Amazonqa: A review-based question answering task. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4996–5002. International Joint Conferences on Artificial Intelligence Organization.

William L. Hamilton. 2020. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159.

Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2019. Beyond English-only reading comprehension: Experiments in zero-shot multilingual transfer for Bulgarian. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 447–459, Varna, Bulgaria. INCOMA Ltd.

Amir Hazem, Basma El Amal Boussaha, and Nicolas Hernandez. 2017a. MappSent at IJCNLP-2017 Task 5: A textual similarity approach applied to multi-choice question answering in examinations. In *Proceedings of the IJCNLP 2017, Shared Tasks*, page 203–207. Asian Federation of Natural Language Processing.

Amir Hazem, Basma El Amel Boussaha, and Nicolas Hernandez. 2017b. MappSent: a textual mapping approach for question-to-question similarity. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 291–300, Varna, Bulgaria. INCOMA Ltd.

Jing He and Decheng Dai. 2011. Summarization of yes/no questions using a feature function model. In *Proceedings of the Asian Conference on Machine Learning*, volume 20 of *Proceedings of Machine Learning Research*, pages 351–366, South Garden Hotels and Resorts, Taoyuan, Taiwain. PMLR.

Qi He, Jaewon Yang, and Baoxu Shi. 2020. Constructing knowledge graph for social networks in a deep and holistic way. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 307–308, New York, NY, USA. Association for Computing Machinery.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th*

International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016, pages 507–517.

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

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, pages 1693–1701, Cambridge, MA, USA. MIT Press.

Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. 2016. WikiReading: A novel large-scale language understanding task over Wikipedia. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1545, Berlin, Germany. Association for Computational Linguistics.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. In *Proceedings of the 4th International Conference on Learning Representations ICLR 2016*, San Juan, Puerto Rico.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

J. J. Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Michael N. Jones and Douglas J. K. Mewhort. 2007. Representing word meaning and order information in a composite holographic lexicon. *Psychological Review*, 114(1):1–37.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin. 2020. *Speech and Language Processing (3rd Edition (Draft))*. Online, USA.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? A critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5010–5015.

Aniruddha Kembhavi, Min Joon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? Textbook question answering for multimodal machine comprehension. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5376–5384.

Tom Kenter, Llion Jones, and Daniel Hewlett. 2018. Byte-level machine reading across morphologically varied languages. In *Proceedings of the The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5820–5827.

Mohamed A. Khamsi and William A. Kirk. 2001. *An Introduction to Metric Spaces and Fixed Point Theory*. John Wiley Sons, Ltd.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 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)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations, ICLR 2017*, ICLR '17.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 3294–3302, 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.

T. Kohonen, P. Lehtiö, J. Rovamo, J. Hyvärinen, K. Bry, and L. Vainio. 1977. A principle of neural associative memory. *Neuroscience*, 2(6):1065–1076.

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.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*,

volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China. PMLR.

Kyungjae Lee, Kyoungho Yoon, Sunghyun Park, and Seung-won Hwang. 2018. Semi-supervised training data generation for multilingual question answering. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 2758–2762, Miyazaki, Japan. European Language Resources Association (ELRA).

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Xiaoxiao Li and João Saúde. 2020. Explain graph neural networks to understand weighted graph features in node classification. In *Machine Learning and Knowledge Extraction*, pages 57–76, Cham. Springer International Publishing.

Xinjian Li, Ran Tian, Ngan L. T. Nguyen, Yusuke Miyao, and Akiko Aizawa. 2013. Question answering system for entrance exams in QA4MRE. In *Working Notes for CLEF 2013 Conference , Valencia, Spain, September 23-26, 2013.*

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2016. Gated graph sequence neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Yichan Liang, Jianheng Li, and Jian Yin. 2019. A new multi-choice reading comprehension dataset for curriculum learning. In *Proceedings of The Eleventh Asian Conference on Machine Learning*, volume 101 of *Proceedings of Machine Learning Research*, pages 742–757, Nagoya, Japan. PMLR.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. KorQuAD1.0: Korean QA dataset for machine reading comprehension. *arXiv:1909.07005*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2829–2839, Hong Kong, China. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Hugo Liu and Push Singh. 2004. Conceptnet — a practical commonsense reasoning toolkit. *BT Technology Journal*, 22(4):211–226.

Shanshan Liu, Xin Zhang, Sheng Zhang, Hui Wang, and Weiming Zhang. 2019a. Neural machine reading comprehension: Methods and trends. *Applied Sciences*, 9(18):3698.

Tzu-Chien Liu, Yu-Hsueh Wu, and Hung-yi Lee. 2017. Query-based attention CNN for text similarity map. *arXiv:1709.05036*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv:1907.11692*.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2020. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *arXiv:2007.15207*, abs/2007.15207.

Hans Peter Luhn. 1957. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development*, 1(4):309–317.

Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. 2017. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1211–1220, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208.

Kevin Lund, Curt Burgess, and Ruth Ann Atchley. 1995. Semantic and associative priming in high-dimensional semantic space. In *Proceedings of the 17th Annual Conference of the Cognitive Science Society*, pages 660–665.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Bill MacCartney and Christopher D. Manning. 2007. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, Prague. Association for Computational Linguistics.

Kateřina Macková and Milan Straka. 2020. Reading comprehension in czech via machine translation and cross-lingual transfer. In *23rd International Conference on Text, Speech and Dialogue*, pages 171–179, Cham, Switzerland. Springer.

Christopher Malon and Bing Bai. 2013. Answer extraction by recursive parse tree descent. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 110–118, Sofia, Bulgaria. Association for Computational Linguistics.

Julian McAuley and Alex Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 625–635.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832, Melbourne, Australia. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas. Association for Computational Linguistics.

George A. Miller and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Ashutosh Modi, Tatjana Anikina, Simon Ostermann, and Manfred Pinkal. 2016. InScript: Narrative texts annotated with script information. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3485–3493, Portorož, Slovenia. European Language Resources Association (ELRA).

Salman Mohammed, Peng Shi, and Jimmy Lin. 2018. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 291–296, New Orleans, Louisiana. Association for Computational Linguistics.

Timo Möller, Julian Risch, and Malte Pietsch. 2021. Germanquad and germandpr: Improving non-english question answering and passage retrieval. *arXiv:2104.12741*, abs/2104.12741.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, Online. Association for Computational Linguistics.

Omar Mossad, Amgad Ahmed, Anandharaju Raju, Hari Karthikeyan, and Zayed Ahmed. 2020. Fat albert: Finding answers in large texts using semantic similarity attention layer based on bert. *arXiv:2009.01004*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. LSDSem 2017 shared task: The story cloze test. In *Proceedings of the*

2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics, LSDSem@EACL 2017, Valencia, Spain, April 3, 2017, pages 46–51.

Sebastian Nagel. 2016. Cc-news. https://commoncrawl.org/2016/10/news-dataset-available/.

Kiet Van Nguyen, Khiem Vinh Tran, Son T. Luu, Anh Gia-Tuan Nguyen, and Ngan Luu-Thuy Nguyen. 2020. A pilot study on multiple choice machine reading comprehension for vietnamese texts. *arXiv:2001.05687*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated MAchine Reading COmprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235, Austin, Texas. Association for Computational Linguistics.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018. MCScript: A novel dataset for assessing machine comprehension using script knowledge. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3567–3574, Miyazaki, Japan. European Language Resources Association (ELRA).

Simon Ostermann, Michael Roth, and Manfred Pinkal. 2019. MCScript2.0: A machine comprehension corpus focused on script events and participants. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 103–117, Minneapolis, Minnesota. Association for Computational Linguistics.

Anusri Pampari, Preethi Raghavan, Jennifer Liang, and Jian Peng. 2018. emrQA: A large corpus for question answering on electronic medical records. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2357–2368, Brussels, Belgium. Association for Computational Linguistics.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Heiko Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508.

Anselmo Peñas, Eduard Hovy, Pamela Forner, Álvaro Rodrigo, Richard Sutcliffe, and Roser Morante. 2013. Qa4mre 2011-2013: Overview of question answering for machine reading evaluation. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 303–320, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. Knowledge graph identification. In *Proceedings of the 12th International Semantic Web Conference - Part I*, ISWC '13, page 542–557, Berlin, Heidelberg. Springer-Verlag.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Boyu Qiu, Xu Chen, Jungang Xu, and Yingfei Sun. 2019a. A survey on neural machine reading comprehension. *arXiv:1906.03824*.

Delai Qiu, Yuanzhe Zhang, Xinwei Feng, Xiangwen Liao, Wenbin Jiang, Yajuan Lyu, Kang Liu, and Jun Zhao. 2019b. Machine reading comprehension using structural knowledge graph-aware network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5896–5901, Hong Kong, China. Association for Computational Linguistics.

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

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

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

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.

Michaela Regneri, Alexander Koller, and Manfred Pinkal. 2010. Learning script knowledge with web experiments. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 979–988, Uppsala, Sweden. Association for Computational Linguistics.

Douglas L. Reilly, Leon N. Cooper, and Charles Elbaum. 1982. A neural model for category learning. *Biol. Cybern.*, 45(1):35–41.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California. Association for Computational Linguistics.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 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*, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.

Stephen E. Robertson and Karen Spärck Jones. 1976. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2021. QA dataset explosion: A taxonomy of NLP resources for question answering and reading comprehension. *arXiv:2017.12708*.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020a. Getting closer to AI complete question answering: A set of prerequisite real tasks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8722–8731. AAAI Press.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020b. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. 2015. A dataset for movie description. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3202–3212.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. In Michael W. Berry and Jacob Kogan, editors, *Text Mining. Applications and Theory*, pages 1–20. John Wiley and Sons, Ltd.

Frank Rosenblatt. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386.

Kalyani Roy, Smit Shah, Nithish Pai, Jaidam Ramtej, Prajit Nadkarni, Jyotirmoy Banerjee, Pawan Goyal, and Surender Kumar. 2020. Using large pretrained language models for answering user queries from product specifications. In *Proceedings of The 3rd Workshop on e-Commerce and NLP*, pages 35–39, Seattle, WA, USA. Association for Computational Linguistics.

Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2087–2097, Brussels, Belgium. Association for Computational Linguistics.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1693, Melbourne, Australia. Association for Computational Linguistics.

Sandip Sarkar, Dipankar Das, and Partha Pakray. 2017. Ju nitm at ijcnlp-2017 task 5: A classification approach for answer selection in multi-choice question answering system. In *Proceedings of the IJCNLP 2017, Shared Tasks*, page 213–216. Asian Federation of Natural Language Processing.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *The Graph Neural Network Model*, 20(1):61–80.

E. W. Schneider. 1973. *Course Modularization Applied: The Interface System and Its Implications For Sequence Control and Data Analysis*. Human Resources Research Organization, Alexandria, VA and Distributed by ERIC Clearinghouse, Washington, D.C.

Guo Shangmin, Liu Kang, He Shizhu, Liu Cao, Zhao Jun, and Wei Zhuoyu. 2017. IJCNLP-2017 Task 5: Multi-choice question answering in examinations. In *Proceedings of the IJCNLP 2017, Shared Tasks*, page 34–40. Asian Federation of Natural Language Processing.

Tatiana Shavrina, Alena Fenogenova, Anton Emelyanov, Denis Shevelev, Ekaterina Artemova, Valentin Malykh, Vladislav Mikhailov, Maria Tikhonova, Andrey Chertok, and

Andrey Evlampiev. 2020. RussianSuperGLUE: A Russian language understanding evaluation benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4717–4726, Online. Association for Computational Linguistics.

Peter Shaw, Philip Massey, Angelica Chen, Francesco Piccinno, and Yasemin Altun. 2019. Generating logical forms from graph representations of text and entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 95–106, Florence, Italy. Association for Computational Linguistics.

Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems*, pages 1223–1237, London, UK. Springer-Verlag.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *16th International Conference on the World Wide Web*, pages 697–706.

Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, volume 28, pages 2440–2448. Curran Associates, Inc.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge dataset and models for dialogue-based reading comprehension. *Transactions of the Association for Computational Linguistics*, 7:217–231.

Simon Šuster and Walter Daelemans. 2018. CliCR: a dataset of clinical case reports for machine reading comprehension. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1551–1563, New Orleans, Louisiana. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q.

Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4631–4640. IEEE Computer Society.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2020. A survey on explainability in machine reading comprehension. *arXiv:2010.00389*.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations, ICLR 2019, Ernest N. Morial Convention Center, New Orleans*.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *arXiv:1806.02847*.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*.

Esko Ukkonen. 1993. Approximate string-matching and the q-gram distance. In *Sequences II*, pages 300–312, New York, NY. Springer New York.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and

R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver Canada, April 30th - May 3rd, 2018*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Mengting Wan and Julian McAuley. 2016. Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems. In *The IEEE International Conference on Data Mining series (ICDM)*, pages 489–498. IEEE.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 3266–3280. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Bingning Wang, Ting Yao, Qi Zhang, Jingfang Xu, and Xiaochuan Wang. 2020a. Reco: A large scale chinese reading comprehension dataset on opinion. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9146–9153.

Chao Wang. 2020. A study of the tasks and models in machine reading comprehension. *arXiv:2001.08635*.

Di Wang, Leonid Boytsov, Jun Araki, Alkesh Patel, Jeff Gee, Zhengzhong Liu, Eric Nyberg, and Teruko Mitamura. 2014. CMU multiple-choice question answering system at NTCIR-11 qa-lab. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-11, National Center of Sciences, Tokyo, Japan, December 9-12, 2014*, pages 542 – 549.

Min Wang, Qingxun Liu, Peng Ding, Yongbin Li, and Xiaobing Zhou. 2017a. Ynudlg at ijcnlp-2017 task 5: A cnn-lstm model with attention for multi-choice question answer-

ing in examinations. In *Proceedings of the IJCNLP 2017, Shared Tasks*, page 194–198. Asian Federation of Natural Language Processing.

Ran Wang, Kun Tao, Dingjie Song, Zhilong Zhang, Xiao Ma, Xi'ao Su, and Xinyu Dai. 2020b. R3: A reading comprehension benchmark requiring reasoning processes. *arXiv:2004.01251*.

Shuohang Wang and Jing Jiang. 2017. A compare-aggregate model for matching text sequences. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017b. 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)*, pages 189–198. Association for Computational Linguistics.

Lilian D. A. Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. 2016. A crowdsourced database of event sequence descriptions for the acquisition of high-quality script knowledge. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3494–3501, Portorož, Slovenia. European Language Resources Association (ELRA).

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Dirk Weissenborn, Tomáš Kočiský, and Chris Dyer. 2017. Dynamic integration of background knowledge in neural NLU systems. *arXiv:1706.02596*.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2016. Towards AI-complete question answering: A set of prerequisite toy tasks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bernard Widrow and Marcian E. Hoff. 1962. *Associative Storage and Retrieval of Digital Information in Networks of Adaptive "Neurons"*, pages 160–160. Springer US, Boston, MA.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*.

Qizhe Xie, Guokun Lai, Zihang Dai, and Eduard Hovy. 2018. Large-scale cloze test dataset created by teachers. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2344–2356, Brussels, Belgium. Association for Computational Linguistics.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. TWEETQA: A social media focused question answering dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5020–5031, Florence, Italy. Association for Computational Linguistics.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. Review conversational reading comprehension. *arXiv:1902.00821*.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1368, Brussels, Belgium. Association for Computational Linguistics.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357, Florence, Italy. Association for Computational Linguistics.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in LSTMs for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446, Vancouver, Canada. Association for Computational Linguistics.

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.

Yinfei Yang, Steve Yuan, Daniel Cer, Sheng-yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018a. Learning semantic

textual similarity from conversations. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 164–174, Melbourne, Australia. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Qian Yu, Wai Lam, and Zihao Wang. 2018. Responding e-commerce product questions via exploiting QA collections and reviews. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2192–2203.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. ReClor: A reading comprehension dataset requiring logical reasoning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Hang Yuan, You Zhang, Jin Wang, and Xuejie Zhang. 2017. Ynu-hpcc at ijcnlp-2017 task 5: Multi-choice question answering in exams using an attention-based lstm model. In *Proceedings of the IJCNLP 2017, Shared Tasks*, page 208–212. Asian Federation of Natural Language Processing.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Changchang Zeng, Shaobo Li, Qin Li, Jie Hu, and Jianjun Hu. 2020. A survey on machine reading comprehension—tasks, evaluation metrics and benchmark datasets. *Applied Sciences*, 10(21).

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018a. ReCoRD: Bridging the gap between human and machine commonsense reading comprehension. *arXiv:1810.12885*.

Xiao Zhang, Ji Wu, Zhiyang He, Xien Liu, and Ying Su. 2018b. Medical exam question answering with large-scale reading comprehension. In *Proceedings of the The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 5706–5713.

Xin Zhang, An Yang, Sujian Li, and Yizhong Wang. 2019. Machine reading comprehension: a literature review. *arXiv:1907.01686*.

Yuhao Zhang, Yuhui Zhang, Peng Qi, Christopher D Manning, and Curtis P Langlotz. 2021. Biomedical and clinical English model packages for the Stanza Python NLP library. *Journal of the American Medical Informatics Association*, 28(9):1892–1899.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. 2018c. Variational reasoning for question answering with knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 6069–6076. AAAI Press.

Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph neural networks: A review of methods and applications. *arXiv:1812.08434*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 19–27, USA. IEEE Computer Society.

Appendix A

Additional Datasets Details

A.1 Additional Features and Statistics of Dataset

Dataset	English Words	Numbers	Not English Words	Not ASCII	Web Links
AmazonQA	76.4%	2.7%	16.8%	0.4%	3.6%
Amazon YesNo	81.3%	2.0%	16.0%	0.0%	0.7%
bAbI	95.4%	0 0%	4.6%	0%	0 0%
BoolQ	75.2%	6.2%	14.4%	4.1%	0.1%
CBTest	88.4%	0.5%	10.9%	0.2%	0 0%
CNN	67.9%	5.7%	24.5%	0.7%	1.3%
CliCR	67.7%	6.4%	25.1%	0.7%	0.1%
CoQA	75.4%	4.4%	17.2%	2.9%	0.2%
CosmosQA	86.0%	2.3%	11.5%	0.0%	0.1%
DREAM	87.8%	7.2%	4.8%	0.1%	0.0%
DROP	61.8%	17.0%	17.0%	4.1%	0.0%
DailyMail	65.9%	7.1%	25.2%	0.7%	1.1%
DuoRC	72.5%	1.2%	22.5%	3.6%	0.0%
emrQA	68.0%	17.3%	14.2%	0.0%	0 0%
HotPotQA	50.2%	4.3%	29.4%	15.8%	0.3%
LAMBADA	70.8%	2.4%	24.4%	1.4%	1.1%
MCScript	95.9%	1.3%	2.5%	0.2%	0.1%
MCScript2	94.4%	1.4%	3.9%	0.2%	0.1%
MCTest 160	95.1%	1.4%	3.3%	0.0%	0 0%
MCTest 500	94.3%	1.0%	4.4%	0.0%	0 0%
MSMARCO	61.6%	7.9%	21.2%	7.4%	2.0%
MovieQA	85.2%	1.8%	13.0%	0.0%	0 0%
MultiRC	84.9%	4.7%	9.6%	0.6%	0.1%
NarrativeQA	79.9%	1.6%	16.0%	2.4%	0.0%

Continued on next page

Table A.1 – *Continued from previous page*

Dataset	**English Words**	**Numbers**	**Not English Words**	**Not ASCII**	**Web Links**
Natural Questions	32.4%	24.5%	20.8%	10.0%	12.2%
NewsQA	72.1%	4.7%	21.3%	0.8%	1.0%
PubMedQA	75.4%	17.1%	6.4%	1.0%	0.0%
QAngoroo MedHop	77.2%	6.3%	14.2%	2.2%	0.0%
QAngoroo WikiHop	57.1%	7.4%	30.9%	4.6%	0.1%
QuAC	72.6%	4.0%	23.2%	0.1%	0.1%
Quasar-S	63.0%	11.1%	21.3%	1 0.0%	4.6%
Quasar-T	55.5%	9.9%	28.3%	1 0.0%	6.3%
RACE-C	79.9%	3.3%	10.4%	6.3%	0.1%
RACE	76.5%	6.4%	16.1%	0.0%	0.9%
ReClor	91.6%	1.8%	6.6%	0.0%	0%
RecipeQA	77.0%	1.6%	16.6%	1.9%	1.3%
SQuAD	66.6%	6.5%	19.2%	7.6%	0.1%
SQuAD2	66.8%	6.5%	19.2%	7.4%	0.1%
SearchQA	60.7%	8.9%	27.3%	0.0%	3.0%
ShaRC	90.6%	5.8%	3.1%	0.3%	0.0%
TriviaQA	44.3%	5.7%	21.2%	24.5%	4.0%
TurkQA	72.1%	3.7%	24.1%	0.0%	0.1%
TyDi	61.8%	3.7%	21.5%	9.6%	0.1%
WhoDidWhat	63.5%	2.1%	34.3%	0.0%	0.0%
WikiMovies	69.0%	0.8%	26.9%	3.3%	0.0%
WikiQA	82.5%	5.2%	9.9%	2.3%	0.1%
WikiReading	38.4%	9.2%	31.1%	20.2%	1.1%

Table A.1: Types of lemmas in dataset vocabulary in percentage.

Dataset	Dataset contains								Dataset statistics					
	Yes No	Non-Factoid	Query	Multi Hop	Multi Doc	Dialogs	No Answer	Extra Data	# instances	# passages	AVG Q_{len}	AVG P_{len}	AVG A_{len}	Vocab Size
AmazonQA	✓	✓	✗	✗	◆	✗	✗	✗	139,905	830,959	16.6	558.2	32.8	1,395,460
AmazonYesNo	✓	✗	✗	◆	✓	✗	◆	✗	40,806	40,806	13.2	4398.2	n/a	864,929
bAbI	✓	✗	✗	✓	✗	✗	✗	✗	20	1,2534	6.3	67.2	1.1	152
BookTest	✗	✗	✗	◆	✗	◆	✗	✗	14,062	14,140,825	-	522	1	1,860,394
BoolQ	✓	✗	✗	◆	✗	✗	✗	✗	8208	12,697	8.8	109.4	n/a	49,117
CBT	✗	✗	✗	✗	✗	✗	✗	✗	108	687,343	30	440	1	53,628
CliCR	✗	✗	✗	✗	✗	✗	✗	✗	11,846	11,846	22.6	1411.7	3.4	122,568
CNN	✗	✗	✗	✗	✗	✗	✗	✗	n/a	107,122	12.8	708.4	1.4	111,198
Daily Mail	✗	✗	✗	✗	✗	✗	✗	✗	n/a	218,017	14.8	854.4	1.5	197,388
Cosmos QA	✗	✓	✗	✓	✗	✗	✗	✗	35,210	35,210	10.6	70.4	8.1	40,067
CoQA	✓	✗	✗	◆	✗	✓	✓	✗	n/a	7,699	6.5	328.0	2.9	59,840
DREAM	◆	✓	✗	✓	✗	✓	✗	✗	6,138	6,444	8.8	86.4	5.3	9,850
DROP	◆	✓	✗	✓	✓	✗	✗	✗	n/a	6147	12.2	246.2	4	44,430
DuoRC	◆	✓	✗	✓	✗	✗	✓	✗	7,477	7,477	8.6	1,260.9	3.1	119,547
emrQA	✓	✓	◆	✓	✗	✗	✓	✓	2427	2,427	7.9	1328.4	2.0	70,837
HotpotQA	✓	✓	✗	✓	✓	✗	✓	✗	534,433	105,257	20.0	1100.7	2.4	741,974
LAMBADA	✗	✗	✗	✗	✗	✗	✗	✗	5,325	10,022	15.4	58.5	1	203,918
MCScript	✓	✓	✗	✗	✗	✗	✗	✗	110	2,119	6.7	196.0	3.6	7,867
MCScript2.0	✗	✓	✗	✗	✗	✗	◆	✗	200	3,487	8.2	164.4	3.4	11.890
MCTest 160	✓	◆	✗	◆	✗	✗	✗	✗	160	160	9.2	241.8	3.7	2,246
MCTest 500	✓	◆	✗	◆	✗	✗	✗	✗	500	500	8.9	251.6	3.8	3,334
MedQA	✗	✓	✗	✓	✓	✗	✗	✗	5	243,712	27.4	4.2	43.2	-
MovieQA	◆	✓	✗	◆	✗	✗	✗	✓	408	408	9.34	727.91	5.6	21,322
MSMARCO	✓	✓	✗	✓	✓	✗	✓	✓	n/a	10,087,677	6.5	65.9	11.1	3,324,030
MultiRC	◆	✓	✗	✓	✓	✗	✗	✗	871	871	4.8	92.4	5.5	23,331
NarrativeQA	✗	✓	✗	✓	◆	✓	✗	✗	1,572	1,572	9.9	673.9	4.8	38,870
NaturalQuestions	✓	✓	✗	✗	✗	✗	✓	✗	109,715	315,203	9.36	7312.13	164.56	3,635,821

Continued on next page

Table A.2 – *Continued from previous page*

Dataset	Dataset contains								Dataset statistics					
	Yes No	Non-Factoid	Query	Multi Hop	Multi Doc	Dialogs	No Answer	Extra Data	# instances	# passages	AVG Q_{len}	AVG P_{len}	AVG A_{len}	Vocab Size
NewsQA	◆	✓	✗	✓	✗	✗	✓	✗	12,744	12,744	7.8	749.2	5.0	90,854
PubMedQA	✓	✗	✗	✓	✗	✗	◆	✗	n/a	3,358	3	15.1	73.8	14,751
QAngaroo WikiHop	✗	✗	✓	✓	✓	✗	✗	✗	n/a	48,867	3.5	1381	1.8	304,322
QAngaroo MedHop	✗	✗	✓	✓	✓	✗	✗	✗	n/a	1962	3	9366.7	1	76,954
QuAC	✓	✓	✗	✓	✗	✓	✓	✗	8853	13,594	5.6	401	14.1	99,912
QuAIL	✗	✓	✗	✓	✗	✗	✓	✗	680	680	9.70	388.29	4.36	17271
Quasar-S	◆	✗	✗	✗	✗	✗	✗	✗	n/a	37,362	24.3	(S)1995.9 (L)5210.1	1.5	(S)660,425 (L)987,380
Quasar-T	✗	✗	✗	✗	✗	✗	✗	✗	n/a	43,012	11.1	(S)2256.2 (L)7372.6	1.9	(S)1,021,823 (L)2,019,336
RACE	✗	✓	✗	✓	✗	✗	✗	✗	n/a	27,933	12.0	329.5	6.3	98,482
RACE-C	✗	✗	✗	✓	✗	✗	✗	✗	n/a	2,708	13.8	423.8	7.4	38,399
Recipe QA	✗	✗	✗	✗	✗	✗	✗	✓	n/a	9,761	10.8	580.0	3.3	62,938
ReClor	✗	✓	✗	✓	✗	✗	✗	✗	n/a	6,138	17.0	73.6	20.6	17,865
ReCoRD	✗	✗	✗	✓	✗	✗	✗	✗	n/a	73190	24.72	193.64	1.5	139724
SciQ	✗	✗	✗	✗	✗	✗	✗	✓	n/a	12,252	14.6	87.1	1.5	23,320
SearchQA	✗	✓	✗	✓	✓	✗	✗	✗	27,995	13,796,295	16.7	58.7	2.1	3,506,501
ShARC	✓	✗	✗	◆	✗	✓	✗	✗	697	24,160	8.6	87.2	4.0	5,231
SQuAD	✗	✓	✗	✗	✗	✗	✗	✗	490	20,963	11.4	137.1	3.5	87,765
SQuAD2.0	✗	✓	✗	✗	✗	✗	✓	✗	477	20,239	11.2	137.0	3.5	88,081
TriviaQA	✗	◆	✗	✓	✓	✗	✓	✗	n/a	801,194	16.4	3867.6	2.3	7,366,586
TurkQA	✓	✓	✗	✗	✗	✗	✗	✗	n/a	13,425	10.3	41.6	2.9	44,677
TweetQA	✗	✓	✗	✗	✗	✗	✗	✗	n/a	13757	8.02	31.93	2.70	32542
TyDi	✓	✗	✗	✗	✗	✗	✓	✓	n/a	14,378	8.3	3,694.2	4.6	848,524
Who Did What	✗	✗	✗	✗	✗	✗	✗	✗	n/a	205,978	31.2	N/A	2.1	347,406
WikiMovies	✗	✓	✓	◆	✓	✗	✗	✓	n/a	186,444	8.7	77.9	6.8	56,893

Continued on next page

Table A.2 – *Continued from previous page*

Dataset	Dataset contains								Dataset statistics					
	Yes No	Non-Factoid	Query	Multi Hop	Multi Doc	Dialogs	No Answer	Extra Data	# instances	# passages	AVG Q_{len}	AVG P_{len}	AVG A_{len}	Vocab Size
WikiQA	✗	✗	✗	✗	✗	✗	✓	✗	n/a	1,242	6.5	252.6	n/a	20,686
WikiReading	✗	✗	✓	✓	✗	✗	✓	✗	4,313,786	18,807,888	2.35	569.0	2.2	8,928,645

Table A.2: Datasets in alphabetical order and additional properties. Where extra data means the English RC task is only one part of bigger dataset with additional resources such as images or video, or there is an availability of resources in other languages. ✓ – present; ◆ – present in a limited form; ✗ – not present.

Dataset	# of Q	what	when	where	which	why	how	who/ whose	how old/ much/ many	boolean	other
AmazonQA	830954	10.2%	0.4%	1.2%	0.4%	0.6%	6.8%	0.1%	2.3%	55.3%	22.8%
AmazonYesNo	80391	0.2%	0.2%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	88.3%	11.1%
bAbI	40000	21.7%	0	36.9%	0	3.1%	5.0%	3.3%	5.0%	25.0%	0
BoolQ	15942	0.1%	0.1%	0.0%	0	0	0.0%	0.0%	0	97.5%	2.3%
CoQA	116630	29.7%	4.1%	6.6%	1.6%	2.7%	4.6%	14.7%	5.4%	20.7%	9.9%
CosmosQA	35210	54.6%	0.2%	1.6%	0.7%	34.2%	5.2%	1.1%	0.3%	1.2%	0.9%
DREAM	9934	56.3%	4.7%	10.0%	2.9%	8.5%	6.5%	3.6%	4.1%	1.1%	2.2%
DROP	86945	6.5%	0.6%	0.5%	18.2%	0.1%	0.7%	8.1%	60.4%	1.7%	3.2%
DuoRC	100966	33.1%	1.0%	8.6%	1.2%	2.5%	3.4%	39.5%	2.6%	1.8%	6.3%
emrQA	1980621	16.1%	0.5%	0.0%	0	1.4%	1.0%	0.0%	0.2%	74.0%	6.8%
HotPotQA	105253	22.6%	2.6%	1.9%	13.5%	0.0%	0.6%	8.6%	1.1%	6.9%	42.0%
MCScript	13939	13.9%	5.8%	9.4%	0.5%	11.6%	13.4%	12.2%	3.8%	28.6%	0.7%
MCScript2	19821	42.0%	27.9%	11.0%	0.2%	0.7%	3.8%	8.4%	0.7%	0.0%	5.2%
MCTest 160	639	51.3%	1.4%	6.9%	2.2%	12.1%	3.9%	13.3%	3.9%	1.6%	3.4%
MCTest 500	2000	52.1%	1.7%	8.5%	3.2%	12.0%	3.5%	12.6%	3.9%	0.8%	1.7%
MSMARCO	1009035	35.6%	2.7%	3.5%	1.8%	1.7%	11.1%	3.4%	5.8%	7.9%	26.6%
MovieQA	29888	46.3%	1.2%	6.8%	0.9%	11.0%	9.4%	19.3%	1.4%	1.9%	1.8%
MultiRC	7903	36.6%	2.3%	3.9%	4.0%	7.0%	6.7%	14.3%	4.6%	7.2%	13.4%
NarrativeQA	46764	38.3%	1.6%	7.5%	2.2%	9.8%	8.3%	24.4%	2.2%	0.1%	5.6%
NaturalQuestions	315104	15.5%	13.1%	10.1%	2.9%	1.2%	2.3%	25.3%	3.8%	2.6%	23.4%
NewsQA	119632	44.3%	4.1%	7.1%	2.2%	0.1%	0.9%	19.8%	5.9%	3.9%	11.7%
PubMedQA	1000	0	0	0	0	0	0	0	0	64.1%	35.9%
QuAC	90922	35.0%	5.2%	3.5%	0.7%	2.8%	6.6%	5.3%	1.4%	36.6%	2.9%
Quasar-S	37362	0.0%	0.0%	0.0%	0	0	0.0%	0.0%	0	1.1%	98.8%
Quasar-T	41102	32.0%	0.5%	2.1%	10.6%	0.2%	0.6%	11.3%	1.4%	0.5%	40.7%

Continued on next page

Table A.3 – *Continued from previous page*

Dataset	**# of Q**	**what**	**when**	**where**	**which**	**why**	**how**	**who/ whose**	**how old/ much/ many**	**boolean**	**other**
RACE-C	11909	17.7%	1.1%	0.3%	10.0%	4.4%	1.4%	0.6%	0.3%	0.1%	64.1%
RACE	51526	35.6%	1.8%	2.3%	23.1%	8.5%	4.3%	2.6%	3.0%	0.4%	18.4%
ReClor	6138	0.2%	0	0	56.5%	0.0%	0.0%	0	0	0.0%	43.2%
SQuAD	98160	43.4%	6.3%	3.8%	4.7%	1.4%	3.3%	9.7%	6.1%	1.2%	20.1%
SQuAD2	142183	46.0%	6.1%	3.6%	4.3%	1.4%	3.2%	9.9%	5.8%	1.0%	18.7%
SearchQA	163981	0.1%	0.7%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.7%	98.3%
ShaRC	24160	0	0	0	0	0	0	0	0	85.4%	14.6%
TriviaQA	800827	18.6%	0.3%	0.8%	19.0%	0.0%	0.3%	9.9%	1.2%	0.3%	49.6%
TurkQA	53700	34.8%	5.7%	6.9%	1.2%%	0.2%	0.9%	6.9%	1.7%	25.7%	16.0%
TyDi	14378	29.0%	20.5%	4.8%	1.4%	0.8%	9.0%	11.8%	13.5%	8.6%	0.6%
WikiMovies	216453	50.6%	1.7%	0.0%	10.4%	0.0%	7.7%	17.3%	0	2.2%	10.1%
WikiQA	1242	54.5%	9.1%	8.9%	0	0	6.5%	13.5%	7.5%	0	0

Table A.3: The percentage of question words per dataset.

A.2 Named Entity Analysis Statistic

Dataset	# all	# unique	# per Q	# per P	no NE %	both %	PER	ORG	LOC	Extra	Date Time % Money	Other Num
AmazonQA	14422460	1450549	0.6	16.7	0.6	0.1	7.4	21.6	2.0	29.1	13.8	19.5
AmazonYesNo	3620420	529106	0.6	106.3	1.2	0.3	6.5	21.8	2.0	29.7	16.0	16.6
bAbI	395215	30	0.6	9.3	0.4	0.6	70.0	3.3	3.3	0	13.3	10.0
BoolQ	183509	55399	0.3	11.2	0.8	0.1	23.6	18.9	8.9	14.1	15.8	8.9
CNN	7845971	236078	1.7	71.6	0.2	0.7	32.9	20.5	7.4	9.7	14.8	7.5
CoQA	240739	68423	0.3	27.3	11.6	2.6	36.3	15.7	11.3	9.4	15.2	6.5
CosmosQA	106000	26945	0.3	2.8	0.8	0.2	26.3	15.1	8.3	11.4	25.5	7.8
DailyMail	18555396	556855	1.9	83.2	0.2	0.8	32.8	18.7	6.6	10.9	19.3	6.7
DREAM	23513	7474	0.3	3.1	1.2	0.3	17.5	7.3	5.7	7.9	46.3	13.0
DROP	389780	70254	1.4	43.7	3.3	8.0	23.8	11.3	9.5	11.8	23.6	17.1
DuoRC	948574	149171	0.9	115.1	4.3	7.1	58.1	11.6	5.7	9.9	6.2	1.6
emrQA	325912	48205	0.1	85.5	-	14.3	15.2	18.9	4.1	5.6	23.5	31.5
HotPotQA	16838268	2207763	2.8	157.2	0.0	0.9	27.5	21.2	7.5	12.9	9.7	3.3
MCScript	7696	2693	0.0	3.5	6.5	0.0	14.7	7.6	6.7	7.4	49.6	11.4
MCScript2.0	11936	3603	0.1	3.1	5.4	0.2	16.3	9.2	5.5	7.6	47.4	9.9
MCTest 160	2998	705	0.8	15.4	1.3	2.6	52.1	4.1	3.5	5.2	27.8	6.2
MCTest 500	8262	1438	0.7	13.6	1.6	2.4	49.0	5.1	5.6	4.4	27.4	5.9
Qangaroo MedHop	615271	38068	0.0	314.4	1.0	0.0	1.8	21.4	3.0	21.6	18.4	31.3
MovieQA	63149	17359	1.6	96.9	2.4	32.9	55.6	11.3	7.9	9.8	8.6	2.0
MultiRC	26006	12783	1.1	25.0	3.9	6.5	32.6	14.9	12.2	9.1	20.3	7.6
NarrativeQA	160196	39319	1.2	66.2	5.1	22.0	57.5	10.1	8.2	10.0	6.9	1.9
NewsQA	917385	160209	0.5	66.9	5.4	3.2	36.3	19.6	7.2	9.0	14.2	7.2
PubMedQA	18123	8069	0.2	5.3	0.8	0.1	1.2	16.5	3.3	4.6	38.8	34.8
QuAC	998843	183212	0.3	77.4	5.5	1.3	32.1	17.4	5.0	8.1	13.6	3.2

Continued on next page

Table A.4 – *Continued from previous page*

Dataset	**# all**	**# unique**	# per Q	# per P	no NE %	both %	**PER**	**ORG**	**LOC**	**Extra**	**Date Time % Money**	**Other Num**
QuAIL	22171	7800	0.6	21.0	9.5	7.3	26.3	14.0	8.3	9.5	30.3	8.4
Quasar-S	3031009	330360	0.2	84.3	0.9	0.1	8.7	39.4	2.1	29.9	4.5	8.7
Quasar-T	1128503	372037	1.0	204.3	0.4	0.5	25.2	19.5	8.3	13.2	11.4	6.0
RACE	501620	127097	0.5	16.3	2.2	1.1	24.7	17.1	7.2	9.7	24.1	10.7
RACE-C	59113	20946	0.7	18.3	2.6	1.4	22.2	17.6	8.9	7.8	27.6	12.4
RecipeQA	186003	28225	0.4	18.7	0.7	0.0	13.8	10.5	2.8	13.1	17.1	34.0
ReClor	15954	8272	0.3	2.3	0.8	0.1	20.5	16.6	11.7	8.6	30.8	9.4
ReCoRD	1965721	285002	1.7	24.3	0.4	0.7	36.3	15.9	6.1	9.8	14.4	5.6
SciQ	24579	7602	0.1	1.9	0.9	0.1	11.8	10.7	7.1	9.3	18.4	39.4
ShaRC	109435	2603	0.2	4.3	0.8	0.2	1.7	18.0	10.5	8.9	49.1	10.1
SQuAD	405442	127630	1.1	14.2	1.3	2.6	21.5	20.7	11.0	16.7	14.8	8.7
SQuAD2.0	436764	128600	1.1	14.1	2.1	3.7	21.3	20.6	11.3	16.9	14.8	8.6
TriviaQA	13469461	1019022	2.5	601.8	0.1	0.7	24.8	18.5	8.0	14.2	12.7	6.2
TurkQA	409638	68629	1.2	6.4	0.1	0.6	32.5	18.6	13.5	11.4	14.7	3.0
TweetQA	55752	13837	0.2	3.9	0.9	0.1	38.2	17.5	11.2	5.2	20.5	3.5
TyDi	3206563	612711	1.0	306.4	0.3	0.5	22.3	18.7	8.4	14.8	13.0	8.1
WhoDidWhat	1023498	216350	5.0	-	-	-	72.1	8.8	2.9	5.6	6.7	2.2
Qangaroo WikiHop	8114649	740465	0.1	165.9	0.9	0.1	25.3	18.6	12.3	12.1	12.3	5.1

Table A.4: Statistics of named entities per dataset reports number of all named entities (**# all**); number of unique named entities (**# unique**) average number of named entities per question (**# per Q**); average number of named entities per passage (**# per P**) percentage of samples without named entities either in question or passage (**no NE %**); and percentage of samples where the same named entity is in both question and passage(**both %**). The percentage of named entities divided buy categories. Where **PER** – person, **ORG** – organisation, **LOC** – geographical location; **Additional** – products, events, languages, work of art, law, nationalities or religious or political groups, buildings, airports, highways, bridges, etc.; **Date / Time / %/ Money** – dates, time, percentage and money; **Other Numerical** – measurements, as of weight or distance, ordinal ("first", "second", etc.), and numerals that do not fall under another type.

					Percentage of Questions						Percentage of Named entities					
Dataset	Do-main	NO NE	#Q	#NE	PER	ORG	GPE LOC	extra	date % $	other	PER	ORG	GPE LOC	extra	date % $	other
Amazon QA	P	515455	830966	531694	2.11	7.79	2.22	7.84	7.16	21.28	3.6	14.77	4.2	15.08	12.92	49.43
Amazon YesNo	P	49323	80391	46431	2.03	9.18	2.4	9.65	4.31	20.07	3.72	17.78	4.74	19.84	8.09	45.84
bAbI	S	16665	40000	24032	58.34	0	0	0	0	0	100	0	0	0	0	0
BoolQ	W+	12205	15942	4312	1.98	0.05	6.84	0.43	3.53	11.94	7.75	0.19	26.58	1.62	13.43	50.44
CNN	N	19739	107122	178050	31.55	20.09	28.46	16.79	19.87	21.03	23.28	14.08	22.37	11.25	13.34	15.68
CoQA	W+	89609	116630	30560	12.13	1.43	3.3	2.01	2.41	3.47	50.15	5.58	13.37	7.95	9.38	13.57
Cosmos QA	S	27415	35210	8952	11.4	1.58	2.11	1.93	4.48	2.13	49.74	6.38	8.62	7.92	18.48	8.86
DailyMail	N	34724	218017	419897	39.41	20.4	26.64	12.68	32.66	19.3	27.49	12.67	19.02	7.36	21.47	12
DREAM	E	7360	10197	3267	14.63	0.64	2.54	1.47	5.66	5.45	49.49	2.02	8.14	4.9	18.24	17.2
DROP	W	20374	86945	120879	24.94	12.55	13.09	15.8	18.17	26.68	23.25	10.55	12.51	15.45	15.68	22.56
DuoRC	W+	31891	100972	88027	60.58	2	3.61	3.24	1.52	2.49	84.78	2.34	4.34	3.84	1.77	2.93
emrQA	E	186035	1979027	124364	2.72	0.68	0.06	0.12	0.52	1.92	43.3	10.81	1.01	1.86	8.41	34.61
HotPotQA	W	2341	105257	290739	54.82	23.56	22.63	49.69	27.33	16.5	29.51	11.38	13.57	26.64	11.8	7.11
MCScript	S	13664	13939	278	0	0.04	0.01	0	0.38	1.54	0	2.16	0.36	0	19.42	78.06
MCScript 2.0	S	18605	19821	1287	1.96	0.34	0.14	0.29	1.23	2.33	31.86	5.21	2.33	4.51	19.27	36.83
MCTest 160	S	210	640	533	62.19	0.94	0.47	0.78	3.28	4.38	88.18	1.13	0.56	0.94	3.94	5.25
MCTest 500	S	779	2000	1477	57.7	0.35	0.8	0.3	1.85	4.1	89.84	0.47	1.08	0.41	2.51	5.69
MovieQA	W	964	14944	23607	87	3.85	6.66	6.04	2.65	3.28	85.17	2.5	4.41	4.08	1.71	2.13
MultiRC	WN	2703	7904	8375	37.18	8.35	14.39	10.77	7.95	10.63	43.65	9.34	16.05	11.93	8.12	10.91

Continued on next page

Table A.5 – *Continued from previous page*

					Percentage of Questions						Percentage of Named entities					
Dataset	Do-main	NO NE	Q	#NE	PER	ORG	GPE LOC	extra	date % $	other	PER	ORG	GPE LOC	extra	date % $	other
Narrative QA	W+	8081	46765	56085	74.14	2.55	6.59	5.06	1.97	3.96	82.57	2.18	5.83	4.38	1.66	3.39
NewsQA	N	68602	119633	65196	18.64	6.13	9.45	6.13	5.21	4.76	37.09	11.84	19.48	11.89	10.14	9.55
PubMed QA	M	814	1000	229	2.1	3.3	4.3	3	2.5	5.8	9.17	17.47	19.65	13.1	12.66	27.95
Qangaroo WikiHop	W	43029	48867	6021	6.27	0.01	0.69	0.14	1.51	3.56	50.97	0.07	5.58	1.13	12.34	29.91
QuAC	W	68631	90922	25703	11.89	2.69	2.53	2.85	4	3.23	44.78	9.63	9.23	10.38	14.33	11.64
QuAIL	S N	6441	12410	7924	33.99	4.89	6.81	4.63	3.18	2.9	61.81	8.18	12.44	7.61	5.16	4.8
Quasar-S	O	32339	37362	6775	0.13	0.66	0.03	0.39	2.07	11.13	0.86	3.72	0.18	2.24	13.85	79.16
Quasar-T	O	14786	43012	46578	24.51	4.46	18.35	20.01	13.97	12.67	27.41	4.35	20.13	20.67	14.08	13.36
RACE	E	61120	97687	46234	20.01	3.03	5.32	5.98	4.51	4.81	46.58	6.64	12.22	13.51	10.29	10.77
RACE-C	E	7096	14122	9630	12.17	4.41	7.11	7.2	4.34	24.82	19.96	7.13	11.97	11.41	6.8	42.72
RecipeQA	O	7098	9761	3434	11.68	3.5	0.68	9.15	0.55	4.74	37.57	11.5	2.13	27.87	1.69	19.25
ReClor	E	4953	6138	1630	7.92	1.06	1.25	0.37	1.04	9.6	47.36	4.23	5.58	1.53	4.48	36.81
ReCoRD	N	26349	110730	184822	32.25	21.43	16.63	12.47	30.2	19.67	24.45	15.36	12.45	8.78	23.39	15.57
SciQ	O	10694	12252	1765	0.47	0.03	0.83	0.05	1.62	10.18	3.29	0.23	5.89	0.34	11.9	78.36
ShaRC	O	18864	24160	5896	0.14	11.18	4.74	2.8	1.29	2.67	0.58	48.52	19.44	11.48	5.29	14.69
SQuAD	W	27942	98169	107994	19.26	13.46	22.92	21.14	13.47	8.9	19.75	13.47	23.68	21.79	12.7	8.61
SQuAD 2.0	W	42750	142192	151290	18.07	13.23	21.69	20.13	14.43	8.81	19.07	13.62	23.12	21.35	14.03	8.81
TriviaQA	O	2986	28458	71970	41.07	10.9	30.24	46.5	31.22	16.28	26.9	5.16	19.93	25.27	14.68	8.06
TurkQA	W	7519	107400	65725	25.87	7.5	6.63	9.43	3.66	2.32	47.13	13.09	12.82	16.71	6.17	4.07
TweetQA	N	11738	13757	2192	7.76	0.09	1.11	0.28	3.63	2.69	49.73	0.59	7.12	1.78	23.36	17.43

Continued on next page

Table A.5 – *Continued from previous page*

					Percentage of Questions						Percentage of Named entities					
Dataset	Do-main	NO NE	Q	#NE	PER	ORG	GPE LOC	extra	date % $	other	PER	ORG	GPE LOC	extra	date % $	other
TyDi	W	3393	13347	12738	19.48	10.93	24.61	20.63	2.99	11.03	22.55	11.86	27.98	22.72	3.2	11.68
WhoDid What	N	388	205978	1023527	96.86	41.55	53.97	54.75	79.43	28.55	25.56	11.51	17.01	14.84	21.72	9.37

Table A.5: Statistics for named entities in the question across datasets, including number of samples without named entities (**NO NE**), total number of questions (**#Q**), total number of named entities in questions (**#NE**), and domain, where **W** – Wikipedia, **W+** – Wikipedia and other resources, **E** – Exams, **M** – Medicine, **P** – Products, **N** – News, **S** – Stories, **O** – other. Percentage of questions containing a particular type of named entities (middle part), and percentage of particular types of named entities in all question per datasets (right part), where **PER** – person, **ORG** – organisation, **GPE LOC** – location and geographical object, **extra** – includes additional named entities such as products, works of art, law, language, etc. **date % $** – includes date, time, percentage, and money related named entities, **other** – includes other numerical named entities such as quantity, ordinal, and cardinal.

					Percentage of Questions						Percentage of Named entities					
Dataset	Do-main	NO NE	#Q	#NE	PER	ORG	GPE LOC	extra	date % $	other	PER	ORG	GPE LOC	extra	date % $	other
Amazon QA	P	132395	12833105	3562935	5.62	12.3	3.07	9.68	14.08	32.8	5.29	15.25	3.45	11.81	15.4	48.8
bAbI	S	51	58	7	6.9	0	0	0	0	5.17	57.14	0	0	0	0	42.86
CBTest	O	37058	44341	7284	14.18	0.19	1.09	0.39	0.31	0.27	86.31	1.14	6.63	2.4	1.89	1.62
CNN	N	3399	23340	20326	59.57	11.44	9.21	4.73	0.6	0.77	68.93	13.25	10.73	5.5	0.7	0.89
CoQA	W+	35181	70033	41837	21.22	5.04	7.73	4.89	7.47	6.55	39.64	9.19	16.72	9.42	13.19	11.84
Cosmos QA	S	77335	94026	19112	7.31	1.76	2.24	2.06	3.41	2.17	40.03	8.94	11.53	10.67	17.54	11.29
DailyMail	N	7544	61404	54662	68.75	9.12	6.73	2.83	0.42	0.58	77.7	10.31	7.65	3.21	0.47	0.66
DREAM	E	18011	23957	6440	5.64	0.72	2.48	2.02	10.8	4.05	22.07	2.69	9.77	8.25	41.49	15.75
DROP	W	5609	20001	20001	30.22	4.71	9.35	8.57	7.15	20.23	38.3	5.27	12.15	11.76	8.29	24.21
DuoRC	W+	32070	71143	47756	41.56	2.92	4.84	3.35	2.45	2.71	73.8	4.51	8.41	5.26	3.79	4.23
emrQA	E	7645	9688	2503	1.18	0.93	0.02	0.02	4.03	15.55	4.59	3.8	0.08	0.08	16.66	74.79
HotPotQA	W	10659	57259	54241	45.54	9.43	10.59	6.04	8.74	5.79	51.28	10.64	14.87	7.13	9.69	6.39
MCScript	S	14923	17461	2671	2.2	0.58	0.85	0.5	6.59	4.03	15.01	4.01	6.29	3.41	43.95	27.33
MCScript 2.0	S	26399	29151	2908	2.2	0.61	0.6	0.41	3.82	1.95	23.45	6.26	6.64	4.13	39.06	20.46
MCTest 160	S	1582	2186	704	19.72	0.32	0.91	0.78	3.25	3.39	71.88	0.99	2.84	3.41	10.09	10.8
MCTest 500	S	4583	6174	1886	19.5	0.4	1.21	0.34	2.66	2.45	76.3	1.33	4.03	1.17	8.91	8.27
MovieQA	W	27818	61409	43807	40.68	3.78	5.63	4.51	2.51	2.44	71.77	5.65	8.59	6.84	3.58	3.57
MultiRC	WN	17459	30977	18838	18.25	5.23	8.77	5.06	6.97	6.37	36.14	10.37	18.35	10.02	12.8	12.32
Narrative QA	W+	38013	75065	45808	34.41	3.06	5.66	4.31	2.51	2.69	67.84	5.26	10.57	7.47	4.22	4.64

Continued on next page

Table A.6 – *Continued from previous page*

					Percentage of Questions						Percentage of Named entities					
Dataset	Do-main	NO NE	Q	#NE	PER	ORG	GPE LOC	extra	date % $	other	PER	ORG	GPE LOC	extra	date % $	other
NewsQA	N	80795	169285	122905	16.44	7.17	12.23	7.96	9.85	9.46	25.23	10.97	21.97	12.03	15.2	14.59
Qangaroo WikiHop	W	11330	13405	2143	4.71	0.06	2.8	0.25	5.57	2.52	29.44	0.37	17.64	1.54	34.86	16.15
QuAC	W	1195	9916	24910	49.24	25.7	17.47	32.08	35.73	20.89	30.23	13.44	9.53	17.67	17.56	11.57
QuAIL	S N	21533	31424	11204	12.45	2.98	5.27	2.78	7.55	2.53	37.58	8.67	16.63	8.1	21.64	7.39
Quasar-S	O	4804	4875	72	0.68	0	0	0	0.04	0.74	45.83	0	0	0	2.78	51.39
Quasar-T	O	21489	25251	3952	7.79	0	1.48	0.06	1.9	3.88	50.23	0.03	10.32	0.43	12.6	26.39
RACE	E	241503	324601	103787	7.31	2.35	5.27	4.78	5.18	4.36	25.34	7.81	18.61	16.37	17.2	14.68
RACE-C	E	41977	51265	11481	2.9	2.27	4.75	3.82	3.34	3.26	14.37	11.12	24.12	18.9	15.7	15.79
RecipeQA	O	5285	6970	1789	16.21	2.15	0.59	2.77	0.47	2.67	64.95	8.78	2.35	11.07	1.9	10.96
ReClor	E	16100	23935	13329	7.15	4.58	6.38	3.55	11.14	8.67	20.06	10.02	14.56	7.86	26.24	21.25
ReCoRD	N	4987	41366	36987	69.54	8.63	6.48	3.08	0.37	0.58	78.29	9.72	7.41	3.52	0.41	0.65
SciQ	O	20139	21110	1006	1.22	0.06	0.34	0.05	0.7	2.26	26.04	1.29	7.85	0.99	14.71	49.11
SearchQA	O	38871	73343	38866	29.98	4.93	6.14	4.67	1.09	1.91	60.43	9.98	14.12	9.21	2.18	4.09
ShaRC	O	694	1158	639	0.52	14.08	9.07	4.06	15.03	4.75	1.41	27.39	21.13	8.92	30.83	10.33
SQuAD	W	33567	72469	50192	17.3	7.72	8.82	8.03	8.93	8.64	27.67	13.04	17.3	14.06	13.82	14.1
SQuAD 2.0	W	32026	69006	47852	17.15	7.65	8.96	8.08	8.95	8.66	27.37	12.93	17.57	14.16	13.83	14.13
TurkQA	O	8856	28875	26467	21.81	10.42	16.54	9.29	12.1	5.55	26.46	13.17	28.48	11.44	13.9	6.53
TweetQA	N	8476	10283	1893	6.54	0.08	1.85	0.28	4.26	4.84	36.03	0.42	10.62	1.58	23.98	27.36
TyDi	W	764	3722	4366	19.8	4.3	11.58	5.94	26.89	19.56	19.7	4.58	19.03	8.89	25.68	22.13
WhoDid What	N	1646	137290	140091	98.01	0.42	0.74	0.3	0.32	0.27	97.74	0.46	0.84	0.32	0.36	0.28

Continued on next page

Table A.6 – *Continued from previous page*

					Percentage of Questions						Percentage of Named entities					
Dataset	Do-main	NO NE	Q	#NE	PER	ORG	GPE LOC	extra	date % $	other	PER	ORG	GPE LOC	extra	date % $	other
Wiki Movies	W	145	154	7	0	0	0	0	0.65	3.9	0	0	0	0	14.29	85.71

Table A.6: Statistics for named entities in the answers across datasets, including number of samples without named entities (**NO NE**), total number of questions (**#Q**), total number of named entities in answers (**#NE**), and domain, where **W** – Wikipedia, **W+** – Wikipedia and other resources, **E** – Exams, **M** – Medicine, **P** – Products, **N** – News, **S** – Stories, **O** – Other. Percentage of answers containing a particular type of named entities (middle part), and percentage of particular types of named entities in all answers per datasets (right part), where **PER** – person, **ORG** – organisation, **GPE LOC** – location and geographical object, **extra** – includes additional named entities such as products, works of art, law, language, etc. **date % $** – includes date, time, percentage, and money related named entities, **misc** – includes other numerical named entities such as quantity, ordinal, and cardinal.

A.3 Complexity Analysis Data

	General Statistics								Method of Creation					Additional Features									Domain					
	#Q	#P	Avg Q	Avg P	Avg A	Vocab	NE	UNE	AG	HG	UG	CRW	KG	Yes No	non fact	query	multi hop	multi doc	dialog	no answ	extra	Wiki	Story	Med	Other	Exam	News
HP	-0.16	-0.11	-0.47	-0.68	0.29	-0.25	-0.18	-0.29	-0.54	0.11	0.07	0.38	0.04	0.36	0.45	0.01	0.51	-0.05	0.25	0.04	0.05	0.13	0.43	-0.41	-0.33	0.03	0.07
SOTA	0.16	0.17	-0.26	-0.46	0.14	-0.06	0.03	-0.02	-0.45	0.30	0.18	0.18	0.14	0.16	0.26	0.10	0.67	0.12	0.25	0.06	-0.26	0.26	0.16	-0.38	-0.36	0.26	0.09
cite	0.15	0.35	-0.17	-0.07	-0.12	0.25	0.22	0.13	-0.03	0.26	-0.15	-0.10	0.08	0.16	0.20	-0.01	0.38	0.22	-0.12	0.04	-0.01	0.16	0.45	-0.28	-0.29	0.02	-0.20
#Tms	-0.11	0.07	-0.27	-0.05	0.50	0.16	0.22	0.20	-0.37	0.16	0.07	0.27	0.16	-0.15	0.24	0.15	0.52	0.25	0.16	0.09	-0.14	0.22	-0.29	-0.26	0.27	0.09	-0.10
#Q	1.00	0.65	0.63	-0.09	-0.36	-0.02	-0.03	-0.04	0.35	-0.12	-0.11	-0.22	-0.08	-0.31	-0.34	-0.06	-0.27	-0.04	-0.09	0.08	-0.14	-0.15	0.16	0.15	-0.16	-0.08	0.14
#P		1.00	0.49	0.21	-0.28	0.70	0.60	0.53	0.15	0.26	-0.08	-0.32	0.37	-0.39	0.01	-0.05	-0.12	0.50	-0.15	0.25	-0.13	0.27	0.07	-0.13	-0.20	-0.11	0.07
Avg Q			1.00	0.16	-0.34	0.12	0.06	0.11	0.67	-0.12	-0.12	-0.51	-0.18	-0.40	-0.38	-0.26	-0.48	-0.12	-0.28	-0.11	-0.12	-0.21	-0.15	0.23	-0.10	-0.15	0.59
Avg P				1.00	-0.21	0.58	0.43	0.56	0.34	0.04	-0.11	-0.31	0.15	-0.22	-0.16	0.04	-0.34	0.26	-0.20	0.00	-0.07	0.11	-0.28	-0.06	0.42	-0.15	-0.17
Avg A					1.00	-0.16	-0.23	-0.28	-0.45	0.43	-0.25	0.06	-0.07	-0.04	0.68	-0.15	0.22	-0.18	0.22	-0.18	0.25	-0.09	-0.13	-0.15	0.19	0.41	-0.22
Vocab						1.00	0.84	0.80	-0.04	0.41	-0.07	-0.25	0.52	-0.27	0.17	-0.03	0.05	0.67	-0.13	0.40	-0.10	0.43	-0.20	-0.09	0.00	-0.09	-0.08
NE							1.00	0.96	-0.10	0.32	-0.08	-0.11	0.73	-0.32	0.05	0.46	0.17	0.94	-0.16	0.30	-0.13	0.61	-0.26	-0.14	-0.11	-0.11	0.01
UNE								1.00	0.01	0.26	-0.08	-0.16	0.67	-0.41	-0.05	0.49	0.09	0.89	-0.19	0.22	-0.16	0.56	-0.35	-0.18	-0.03	-0.10	0.12
AG									1.00	-0.36	-0.16	-0.63	-0.30	-0.38	-0.63	-0.16	-0.61	-0.24	-0.30	-0.36	0.09	-0.36	-0.04	0.09	0.17	-0.24	0.42
HG										1.00	-0.10	-0.39	0.16	-0.00	0.57	-0.10	0.36	0.26	0.16	0.08	0.26	0.39	-0.29	-0.15	-0.29	0.67	-0.15
UG											1.00	0.26	-0.09	0.22	-0.18	-0.05	0.16	-0.07	-0.09	-0.10	-0.07	0.46	-0.13	-0.07	-0.13	-0.07	-0.07
CRW												1.00	-0.06	0.28	0.25	0.26	0.24	0.06	-0.06	0.33	-0.26	0.09	0.32	0.06	-0.09	-0.26	-0.26
KG													1.00	-0.13	-0.06	0.55	0.30	0.80	0.23	0.16	-0.13	0.50	-0.24	-0.13	0.05	-0.13	-0.13
YesNo														1.00	0.09	-0.22	0.38	-0.32	0.40	-0.00	0.00	-0.00	0.20	0.00	0.00	-0.00	-0.32
non fact															1.00	-0.18	0.24	0.06	-0.06	0.09	0.06	0.09	0.32	-0.26	-0.30	0.38	-0.26
query																1.00	0.16	0.69	-0.09	-0.10	-0.07	0.46	-0.13	-0.07	-0.13	-0.07	-0.07
m-hop																	1.00	0.24	0.30	0.11	-0.09	0.36	-0.17	-0.09	-0.17	0.24	-0.09
m-doc																		1.00	-0.13	0.26	-0.10	0.67	-0.19	-0.10	-0.19	-0.10	-0.10
dialog																			1.00	0.16	-0.13	-0.19	-0.24	-0.13	0.35	0.34	-0.13
no answ																				1.00	-0.15	0.08	-0.02	0.26	-0.02	-0.15	-0.15
extra																					1.00	0.26	-0.19	-0.10	0.16	-0.10	-0.10
Wiki																						1.00	-0.29	-0.15	-0.29	-0.15	-0.15
Story																							1.00	-0.19	-0.38	-0.19	-0.19
Med																								1.00	-0.19	-0.10	-0.10
Other																									1.00	-0.19	-0.19
Exam																										1.00	-0.10
News																											1.00

Table A.7: Pearson Correlation between Datasets' properties. Where positive strong correlation ($r > 0.5$) with green cell; positive medium correlation($0.3 < r < 0.5$) with yellow cell; negative medium correlation ($-0.5 < r < -0.3$) with orange cell; and finally negative strong correlation ($r < -0.5$) with red cell.

A.4 Other Datasets

There are a number of datasets I did not include in my analysis in Chapter 2 as I focus on the Question Answering Machine Reading Comprehension task. In this section I mention some of these and explain why they are excluded.

A.4.1 Question Answering Datasets

CLOTH (Xie et al., 2018) and **Story Cloze Test** (Mostafazadeh et al., 2016, 2017) are cloze-style datasets with a word missing from the context and without a specific query. In contrast, cloze question answering datasets considered in this work have a passage and a separate sentence which can be treated as question with a missing word.

As well as this, I did not include a number of MRC datasets where the story should be completed such as **ROCStories** (Mostafazadeh et al., 2016), **CODAH** (Chen et al., 2019), **SWAG** (Zellers et al., 2018), and **HellaSWAG** (Zellers et al., 2019) because there are no questions.

QBLink (Elgohary et al., 2018) is technically a MRC QA dataset but for every question there is only the name of a wiki page available. The "lead in" information is not enough to answer the question without additional resources. In other words, QBLink is a more general QA dataset, like CommonSenseQA Talmor et al. (2019).

Another general dataset is **MKQA** Longpre et al. (2020) which contains 260k question-answer pairs in 26 typologically diverse languages.

Textbook Question Answering (**TQA**) (Kembhavi et al., 2017) is a multi-modal dataset requiring not only text understanding but also picture processing.

MCQA is a Multiple Choice Question Answering dataset in English and Chinese based on examination questions introduced as a Shared Task in IJCNLP 2017 by Shangmin et al. (2017). The authors do not provide any supportive documents which can be considered as a passage so it is not a reading comprehension task. More details of this dataset were presented in Chapter 4 as I worked with it.

A number of datasets, such as **SimpleQuestions** (Bordes et al., 2015) and **WebQues-**

tion (Berant et al., 2013), were created with the idea of extracting answers from a knowledge graph. Even though additional resources are involved, it is presented in a structured form rather then a natural text so I do not consider those datasets in the current chapter.

A.4.2 Non-English Datasets

While I focus on English datasets, there are a growing number of MRC datasets in other languages. In this section I will briefly mention some of them. Please see Rogers et al. (2021) for a more complete list.

A.4.2.1 Chinese datasets

DuReader (He et al., 2018) is a Chinese RC dataset. It contains mixed types of questions based on Baidu Search and Baidu Zhidao.[1]

ReCO (Wang et al., 2020a) **Re**ading **C**omprehension dataset on **O**pinion) is the largest human-curated Chinese reading comprehension dataset containing 300k questions with *"Yes/No/Unclear"* answers.

CLUE benchmark Xu et al. (2020) contains a number of Chinese reading comprehension tasks with different difficulty.

A.4.2.2 Other Languages

The extended version of **WikiReading** (Kenter et al., 2018) apart from 18M English questions also contains approximately 5M Russian and about 600K Turkish examples.

TyDi (Clark et al., 2020) is a question answering corpus of 11 Typologically Diverse languages (Arabic, Bengali, Korean, Russian, Telugu, Thai, Finnish, Indonesian, Kiswahili, Japanese, and English). It contains 200k+ question answers pairs based on the Wikipedia articles in those languages.

MLQA (Lewis et al., 2020) contains over 12K question-answer pairs in English and 5K in each of the 6 following languages: Arabic, German, Spanish, Hindi, Vietnamese

[1] `zhidao.baidu.com` – last verified June 2021

and Simplified Chinese, with each question-answer instance parallel between 4 other languages on average.

ViMMRC Nguyen et al. (2020) is multiple-choice questions RC dataset in Vietnamese language. It contains 2,783 questions based on a set of 417 texts.

Hardalov et al. (2019) created an exam-and-quiz-based dataset for Bulgarian, containing 2,636 multiple-choice questions with additionally extracted context from variety of topics (biology, philosophy, geography, and history).

The **RussianSuperGLUE** benchmark (Shavrina et al., 2020) contains several reading comprehension datasets: **DaNetQA** (Glushkova et al., 2020) contains 2.7k boolean questions, **MuSeRC** and **RuCoS** (Fenogenova et al., 2020) are two datasets with 5k and 87k questions, which require reasoning over multiple sentences and commonsense knowledge to infer the answer.

A number of datasets have been created following the approach of SQuAD: **FQuAD** (d'Hoffschmidt et al., 2020) is a French Native Reading Comprehension dataset with 25,000+ questions; **KorQuAD** (Lim et al., 2019) has 70,000 original questions in Korean; The Russian **SberQuAD** (Efimov et al., 2020) dataset contains about 90K examples. **GermanQuAD** (Möller et al., 2021) is a German dataset, as the name suggests. It contains almost 14k extractive questions. All four datasets are based on Wikipedia.

A.4.2.3 Dataset Translation

SQuAD has been semi-automatically translated into several other languages as: Korean **K-QuAD** (Lee et al., 2018); Italian **SQuAD-it** (Croce et al., 2018); Japanese and French (Asai et al., 2018); Spanish **SQuAD-es** (Carrino et al., 2020); Hindi (Gupta et al., 2019a); and Czech (Macková and Straka, 2020).

Artetxe et al. (2020) presented a new multilingual dataset **XQuAD** by translating SQuAD into ten languages (Spanish, German, Greek, Russian, Turkish, Arabic, Vietnamese, Thai, Chinese, Hindi) with professional translators. It contains 240 paragraphs and 1190 question-answer pairs in each language.

Appendix B

Error Analysis Details

B.1 MovieQA Example

Example (B.1) shows extracted sentences which do not contain answer to the question.

(B.1) **Movie:** "2 Guns", 2013 (IMDB ID:*tt1272878*)

Question: What threat does Greco hold over Trench and Stigman if they don't return his money?

Plot Synopses: After beating them and receiving a visit from Earl, *Greco gives the pair 24 hours to steal the money from the Navy and return it to him, or Deb will die*.

Extracted sentences:[1] After the heist, Stigman follows orders to betray Trench and escape with the money, managing to pull his gun right as Trench is about to pull his own. Unknown to Stigman, Trench is an undercover DEA agent and reports to his superior, Jessup (Robert John Burke), that he failed to acquire cocaine from Greco that they could use as evidence to convict him. Unwilling to kill Trench, Stigman wounds Trench in the shoulder and leaves with the money. Against Jessup's orders, Trench decides to remain undercover and assist Stigman in robbing $3 million from Greco, so they can prosecute Greco for money laundering. Trench and Stigman kidnap Greco and interrogate him in the garage at Deb's house, where they find out Earl, Greco's associate, is a black ops operative, and they have stolen

[1] I removed duplicates of the sentences if they were extracted by different methods

money from the CIA. During a standoff among Quince, Earl, Trench, and Stigman, Earl reveals that the CIA has 20 other secret banks, and the loss of the $43.125 million is only a minor setback. He later realizes that the money is in a motel room that he and Deb frequented and goes to help Stigman, who had returned to Greco's farm alone to exact vengeance. Trench and Stigman kill Greco and the duo escapes, but not before Trench shoots Stigman in the leg as payback for shooting him in the desert. Trench has a rendezvous with fellow DEA agent and former lover, Deb Rees (Paula Patton), who is also seeing another man, while Stigman, an undercover Intelligence Specialist with the Navy SEALs, meets with his commanding officer, Harold Quince (James Marsden), who instructs Stigman to kill Trench so the Navy can use the stolen money to fund unauthorized covert operations. He later realizes that the money is in a motel room that he and Deb frequented and goes to help Stigman, who had returned to Greco's farm alone to exact vengeance. While planning to continue to take down the CIA's secret banks and sabotage their black operations, Trench reveals to Stigman that he did not blow up all the money and had some stashed away.

Predicted Answer: That Teemo will die.

Correct Answer: That Deb will die.

Explanation: The sentence which supports the correct answer was not extracted.

B.2 AmazonYesNo

	Work		*Come whit*		*Use*	
	No	**Yes**	**No**	**Yes**	**No**	**Yes**
Correct	14	214	70	23	45	115
Errors	113	8	7	39	57	18
Mixed	103	135	59	62	86	89

Table B.1: Count of questions with answer "Yes" and "No" across stable correct (errors) and mixed samples.