Common Sense Q&A model

Kusma Thummagunta[1] , **Kalyan Naidu Mullaguri**[2]
CAP 6307

Abstract

"*It's just Common sense!*" On an average day, the word common sense is used by at least one person in a group of five. Common sense is defined as sound and practical judgement in every day activities. For example, "*Why does a normal human stay away from fire? - It burns,common sense!*" So common sense is a knowledge that is acquired imperceptibly from the time a species is born. It is developed in humans through life experiences by reflecting on situations before making decisions. There's been a lot of research work on this particular area in recent times on machines to adopt reasoning ability which humans possess. In contrast to the conventional Q&A, Common sense question-answering task is extremely challenging as it is based on the extensive knowledge of the data in the real world and there are very few commonsense reasoning datasets available that can help in testing the state of art models. For this task, we trained a dataset called riddlesense which was developed for answering riddles using one of the best transformer models. Many models were studied for this task and the best of all models is implemented due to resource constraints.

1 Introduction

In contrast to the domain knowledge Q&A, answering a common sense question requires extensive knowledge of the real world along with good understanding of semantics. This is easy for a human with reasoning abilities but for a machine, it is however a challenging task as there is no well-written data. Humans use the prior knowledge on spatial relations, causes and effects, facts related to science and other social conventions when they try to answer a question. For instance,*"The computer which I had just put into the machine room on the fifth floor crashed."* Now, if asked a question on what got crashed, a human can infer that computer is crashed and not the floor. This kind of inference is trivial yet answerable for a human brain but it for a machine it is still out of reach. For machines, more sophisticated techniques of reasoning are required to answer questions just like a human. Currently there's been a huge amount of reach work going on towards these Natural Language processing and Language models with dependencies. Commonsense Q&A is one of the important one in training a model to understand these inferences and dependencies.

Common Sense Q&A involves answering by making use of inferences and comprehensive knowledge in addition to good sense of the question understanding. Usually, common sense questions doesn't clear express all these dependencies and inferences. In order to answer such unusual questions, the machine should predict the correct answer by reasoning with the common sense inferred in the question. Unlike humans, it is hard for the machine to deduce such dependencies and relationships between knowledge.Even, the high performing pre-trained language models of NLP needs additional work in order to solve the common sense Q&A. Selecting the appropriate state of art model that can predict the correct answer out of all the choices for a given question. Various models have been studied in this process and various datasets have been analyzed. Finally, to achieve the prediction accuracies, we selected RiddleSense Dataset that is developed by (Bill Yuchen Lin, 2021) and the language model used is T5 transformer model by(Colin Raffel, 2020).

2 Question Answering

Question-answering(Q&A) is a popular task where given a query, a response is generated based on the information provided with it. Q&A can be broadly classified into three basic types categorized based on the form of the answer. They are : Span-based Q&A, Multiple choice Q&A, and Generative Q&A.

In Q&A field, a substantial amount of research was done on span-based Q&A. For a given question and a specific scenario, the target of this task asks the machine to retrieve a specific text as the answer from the given scenario. Not that the scenario always holds the answer, the given scenario wouldn't have the actual answer sometimes, but it contains the content from which the answer can be achieved. For span-based Q&A, there are so many datasets availabe and few important datasets are : SQuAD (Rajpurkar et al., 2016),TriviaQA (Joshi et al., 2017), and NewsQA (Trischler et al., 2017), and as well as many models can be used such as BiDAF (Seo et al., 2016), DocQA (Yang et al., 2019a), and BERT (J. Devlin and Toutanova, 2018)(J.Devlin et al.,2019) shows high performance.

In multiple-choice Q&A, a question with multiple options is provided from which a option is selected as the answer. Unlike span-based Q&A, context is not mandatory for this type of questions. Typical datasets include the MCTest(Richardson et al., 2013), RACE (Lai et al., 2017), CommonsenseQA (Talmor et al.), and OpenbookQA (Mihaylov et al., 2018).

Finally, generative Q&A is a process in which the answer is obtained from the given question and the context of the question with the help of logical thinking. This is by far the most relevant type of Q&A for real world applications since there is not limitations based on context or the options. However, it is the most complicated among all the QA tasks. The typical datasets for generative Q&A are MS MARCO (Nguyen et al., 2016), SearchQA (Dunn et al., 2017), and NarrativeQA (Kociský et al., 2017).

3 Related Work

The importance of knowledge on reasoning ability and machine common sense about the real world has been acknowledged many decades ago as a critical component in understanding the natural language. Early work dates back to (McCarthy, 1959) - the attempts to find the programs that could help in reasoning about the environment in natural language. (Winograd, 1972)- leveraged a world model for understanding the language in great depth. There's been a little progress in developing common sense representations and exploring the procedures of inference (McCarthy and Hayes, 1969; Kowalski and Sergot, 1985). The significant work happened in the last decade. Works related to Q&A was mainly focused on the fact based questions which can be answered using the context and didn't require common sense (Hermann et al., 2015; Rajpurkar et al., 2016; Joshi et al., 2017; Nguyen et al., 2016; Hermann et al., 2015).

The common sense concept was directly targeted by Winograd Scheme Challenge (Levesque, 2011) is one important benchmark that asked the models to solve paired instances of coreference resolution accurately. COPA(Roemmele et al., 2011) is equally important and for each question, it asks for two best alternatives that reflects the cause or effect to the premise. Though both of the datasets were tough and with high difficulty in generating examples, scalability is the issue during evaluation of new models. Crowd sourcing in latest years helped in the development of many large datasets that focuses on predicting the relations in knowledge and situations, make inferences and connects the events of natural language.

SWAG (Zellers et al., 2018) is most significant development where given an event's textual description, it asks the models to choose what is the probable subsequent event is after the initial event. By fine tuning a pretrained LM on the target tasks, a high performance is achieved on SWAG but there is always the difficulty in setting the benchmarks that measure the understanding of the program.

3.1 Datasets

Every dataset is created for different purposes and choosing the appropriate dataset was the major task in this project. Finally, after going through a huge list of all the significant datasets, we found that both RiddleSense dataset(Bill Yuchen Lin, 2021) and Commonsense QA(CSQA)(Talmor et al.) target general common sense knowledge through Q&A task. Though both of them have the same format, CSQA is more focused on questions that are straight forward where the answering concept's description is easy to understand while RiddleSense use riddles as questions so as to test the high order commonsense reasoning ability.Therefore, RiddleSense dataset is used to assess the performance of the T5 model used.

3.2 Models

There are various Language models that are available for Q&A task. Examples : BERT, ALBERT, XLNet, RoBERTa, ConvBERT and BART.(Pearce et al., 2021) - All the mentioned transformer models are compared and evaluated using different

Q&A datasets and the results show that RoBERTa and BART models work best across all datasets and BERT-BiLSTM outperforms the base BERT model. Various differences in the performances are analyzed between the pre-trained models which are fine tuned on Q&A datasets with different levels of difficulty. It requires lot of resources in order to implement all these models on different datasets. It consumes a lot of memory which was a a major obstacle in our implementation process and hence, using the information and knowledge gained from the paper, we restricted the evaluation process to only one model - T5.

In NLP, transfer leraning emerged as the robust technique and using this, new framework was introduced (Colin Raffel, 2020) that converts all text-based language problems into text-to-text format. T5 (Text-To-Text Transfer Transformer) is an encoder-decoder model. It is pre-trained on the multitask pool of supervised and unsupervised tasks where each task is converted into text-to-text format. Every task like classification, Question-Answering,Translation, Summarization etc is given as the model's input and some target text is generated by training it. T5 uses relative scalar embeddings and it comes in various sizes : T5-small, base, large, 3b, 11b and there are other follow-up works like T5v1.1, mT5, byT5 etc. Selecting T5 models to train the RiddleSense dataset is the best choice considering it is one of the powerful transformer model in recent years and to get hands-on with it is quite a challenging task. Since, the other basic models have been worked on, using that research as a study, we decided to work on T5 model both T5base and T5small to test how this performs on the Q&A tasks for different datasets.

4 Implementation

4.1 Dataset

The first and most important task involved in any machine learning problem is collecting the data that is appropriate to our problem. So, let's talk about the dataset we used for our problem. When we were searching for the right data, that involves a common sense question, choices of answer and the real answer, we came across the riddlesense dataset. This data may seem like it is for a whole different problem but, no. This dataset contains very good questions which are riddles but they are so easy that they are just simple common questions. Example question from the dataset, *What chins are never shaved? choices: A. sea urchins, B. depilation, C. contract, D. mentum, E. Free trade zones* and the answer is choice *A. sea urchins.* Let's see another one, *Why does this light shine? choices: A. pitch black, B. it is sunny, C. flasher, D. it was turned on, E.illumination* and the answer is choice *D. it was turned on.* From the given examples one thing we can clearly observe is that the questions are pretty easy and the choices are just some irrelevant words with only few are close to the answer. So, more than the question, choices make the questions much easier.

Let's see more about what all this dataset contains. When downloaded, it was in .jsonl (json line) format. It has four fields id, questions with the real question inside the stem field, choices with labels and text, and the answer key which is the right answer for that question. We used three dataset files csqa_train.jsonl with 9740 different questions, rs_train with 3510 different questions, rs_dev.jsonl with 1020 questions. Here, we used csqa_train.jsonl and ra_train.jsonl for training, and rs_dev.jsonl for testing.

4.2 Model

We decided to use T5 model for our problem because, the T5 model is best suited for the data we gathered to use for this problem. We say that because for this model and on our data, the preprocessing steps are not very complicated. We also chose this model because it is not only one single model but 5 models t5-small, t5-base, t5-large, t5-3b, t5-11b. These are built on different complexities. Since we have very limited resources, we used t5-small and t5-base for our problem.

4.3 Preprocessing

Since we have our data in json line format, we have to transform it the form which our model accepts. We also remove unnecessary fields and modify few fields in this step.

The model takes a dataframe as an input and it has to have specifically three columns, prefix, input_text, target_text. Prefix is to give the model what exact task is it training, input_text is to showto our model that these are the inputs, and target_text is the desired output. To transform our data into this desired format, we did write a preprocessing code.

In this preprocessing code, we first open this files in read mode and stored each line. We extracted each field from the lines and stored into respective

lists. These lists are loaded as columns into a pandas dataframe with renamed columns as input_text, choices, target_text. In this step we had to omit 'id' field since we have no use of it. Since the answerkeys were just labels but not text, we replaced those labels with the text in the dataframe. Now, as we need our model to be trained on the choices as well, we need to send them to the model through input_text. For this, I did joined the input_text column which has only questions now, with choices column with '#' as a separator between the question and choices as well. Once this is done, we added a new column 'prefix' to our dataframe at the first position. We set this to a constant value 'answer question'. Then, we omit the 'choices' column from our dataframe. With this our dataframe is ready in required format with only three columns (prefix, input_text, target_text).

All the above preprocessing steps are defined inside the read_file() function to provide re-usability. We send the train file to the function call and it returns the train dataframe, same is done to test file as well using the function call. Our dataframes are ready for training and testing the model.

4.4 Training

As we have obtained our dataframes, they are fed to the model for training. We trained t5-small model on rs_train and csqa_train data, and t5-base model on rs_train data.

Due to lack of resources, we couldn't use bigger models, t5-base was also not trained on csqa_train because of less memory and the program execution crashes. So, this is the limit where our systems could handle. To give an example on the time required to run a bigger model, T5-large will take somewhere around 12 hours to train on a very good GPU.

For training, we use 'train_model()' function. We pass our train dataframe to this function along with arguments and parameters. The parameter passed is 'use_cuda=False', if it is set to 'True', then it is a signal to the trainer to use the GPU instead. If not, it'll run on CPU. Some of the arguments passed to the train_model are,

- max_seq_length: Samples are truncated upon the value. Shorter the better.
- train_batch_size: Bigger the better depending on the GPU or CPU.
- eval_batch_size: Same as train_batch_size.
- num_train_epochs: More the number better the model but, also increases the training time.
- evaluate_during_training: Periodically tests the model to check how well it is learning.
- evaluate_during_training_steps: Same as train_batch_size.

Once, the model is done training it'll give loss values for train and test data.

4.5 Evaluation

T5 model has a function called 'eval_model()' to evaluate the model. It'll return a dictionary containing the evaluation results. Unfortunately, due to shortage in memory, eval_method() crashed the execution and did not work. Train method used almost all of the available memory.

But, we wrote a few lines of code to manually calculate the accuracy. We used 'predict()' method to predict for every question from the test dataframe and store all the predictions from model. Then, we compared each output from model to target_text from the test dataframe and counted how many were matching and did a simple division between count and length of the output array that store all the predictions. This gave the accuracy.

5 Results

Results of the models are discussed here. First, T5-small, when first trained on rs_train data which has slightly difficult questions compared to csqa_train but smaller in size, and upon evaluating (tested) on rs_dev data, which is slightly difficult, it scored about 23% accuracy and when trained on csqa_train which is slightly easy and bigger in size and tested on rs_dev which is slightly difficult, it scored about 18%. Next, T5-base, when first trained on rs_train and testing on rs_dev gave the best accuracy so far about 34%. Next when we tried to train on csqa_train data, the program execution has crashed because of shortage in memory. This was a big file and T5-base has more parameters than T5-small. Due to lack of resources, training T5-base on csqa_train was halted. These accuracy values can be seen in the table 1.

MODEL	RS_TRAIN	CSQA_TRAIN
T5-SMALL	23%	18%
T5-BASE	34%	-

Table 1: T5 model accuracies

Model	RS_TRAIN		CSQA_TRAIN	
	Train	Eval	Train	Eval
T5-SMALL	0.08	2.35	1.48	1.95
T5-BASE	0.66	1.47	-	-

Table 2: Model losses

Model losses on training and evaluation can be seen in the table 2.

Conclusion

In this paper, the multi-functional transfer Transformer models T5 - base and T5- small are evaluated on the commonsense Q&A tasks. These state of art language models are fine tuned on the RiddleSense dataset which is a latest benchmark for the comprehensive CommonSense analysis tasks. The experimental results shows that T5- Base model achieved a best of 35% accuracy on the RS dataset and works better than T5-small model. The models trained on less difficult data and tested on slightly difficult data gave an accuracy of 18% which is a quite good score for the number of documents it is trained on, this shows how good T5 model is. Though the accuracies obtained are way lower than the human accuracies, these experiments demonstrate that there is a scope for continuing the research work in this field.

Future Work

Both T5 and RiddleSense datasets are introduced last year and working on the recent developments increases the potential scope of the future work. Our main interest in the further research on the T5 model involves using all the available T5 models like T5-large, T5-11b which has 11 billion parameters. We also want to include comparison between T5 and few huggingface transformer models like BERT models. These can be achieved with good resources such as a virtual desktop with large memory or a very good GPU because these models will take days to train.

Individual Contribution

Both the authors have their parts in all the tasks involved in this project. But few things like literature survey involving study in numerous papers in this area and finding all the important models that can be used are majorly handled by Author 1. Other few tasks like preprocessing and training are majorly handled by Author 2. All other tasks including the reporting works are shared equally.

References

Yichi Yang Dong-Ho Lee Xiang Ren Bill Yuchen Lin, Ziyi Wu. 2021. Riddlesense: Answering riddle questions as commonsense reasoning. *CoRR*, abs/2101.00376.

Adam Roberts Katherine Lee Sharan Narang Michael Matena Yanqi Zhou Wei Li Peter J. Liu Colin Raffel, Noam Shazeer. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research 21 (2020)*, abs/1910.10683.

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. *CoRR*, abs/1704.05179.

Joshua Feldman, Joe Davison, and Alexander M. Rush. 2019. Commonsense knowledge mining from pretrained models. *CoRR*, abs/1909.00505.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

K. Lee J. Devlin, M.-W. Chang and K. Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *Computing and Language Research Repository*, abs/1810.04805.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *CoRR*, abs/1705.03551.

Tomás Kociský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2017. The narrativeqa reading comprehension challenge. *CoRR*, abs/1712.07040.

Robert Kowalski and Marek Sergot. 1985. A logic-based calculus of events. *New Generation Computing*, 4:67–95.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: large-scale reading comprehension dataset from examinations. *CoRR*, abs/1704.04683.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Hector Levesque. 2011. The winograd schema challenge.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *13th International Conference on the Principles of Knowledge Representation and Reasoning, KR 2012*, Proceedings of the International Conference on Knowledge Representation and Reasoning, pages 552–561. Institute of Electrical and Electronics Engineers Inc. 13th International Conference on the Principles of Knowledge Representation and Reasoning, KR 2012 ; Conference date: 10-06-2012 Through 14-06-2012.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

John McCarthy. 1959. Programs with common sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, pages 75–91, London. Her Majesty's Stationary Office.

John McCarthy and Patrick Hayes. 1969. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and Donald Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

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.

Kate Pearce, Tiffany Zhan, Aneesh Komanduri, and Justin Zhijun Zhan. 2021. A comparative study of transformer-based language models on extractive question answering. *ArXiv*, abs/2110.03142.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.

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.

Melissa Roemmele, Cosmin Bejan, and Andrew Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603.

Jingjing Xu Duyu Tang Nan Duan Ming Gong Linjun Shou Daxin Jiang Guihong Cao Shangwen Lv, Daya Guo and Songlin Hu. 2020. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, USA*, pages 8449–8456. AAAI Press.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. *CoRR*, abs/1612.03975.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. "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)*.

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.

Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2015. Machine comprehension with syntax, frames, and semantics. pages 700–706.

Terry Winograd. 1972. Understanding natural language.

Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019a. Simple and effective text matching with richer alignment features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4699–4709, Florence, Italy. Association for Computational Linguistics.

Yunyeong Yang and Sangwoo Kang. 2020a. Common sense-based reasoning using external knowledge for question answering. *IEEE Access*, 8:227185–227192.

Yunyeong Yang and Sangwoo Kang. 2020b. Common sense-based reasoning using external knowledge for question answering. *IEEE Access*, 8:1–1.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

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.