4-20-2021

Evaluation of state-of-the-art NLP deep learning architectures on commonsense reasoning task

Guo Rui (Justin) Lee
Western Michigan University, justinlee38@outlook.com

CS 4910 – Software and System Design II: Implementation and Testing

Project Final Report

Team Member:

Anson Zeeping Lim

Justin Lee

Nathan Brackenbury

ThongSheng Ang

0. Abstract

The goal of this project was to explore modern neural network technology in the application of discerning and generating statements that are ‘reasonable’, in what is known as commonsense reasoning. We built off of the work of Saeedi et al. In their work on the 2020 SemEval task, Commonsense Validation and Explanation (ComVE). SemEval is a workshop that creates a variety of semantic evaluation tasks to examine the state of the art in the practical application of natural language processing. This particular task involved three sections: task A, Validation, in which a program tries to select which of two statements is more sensical; task B, Explanation, in which the program is given an illogical statement and has to choose between three statements to find the one that works as the best explanation as to why the statement is illogical; and task C, generation, in which the program must generate a novel explanation as to why a given statement is illogical.

The existing project that we had taken on to improve had already had considerable success with tasks A and B using a relatively straightforward application of the huggingface transformer library, a python library that acts as a useful wrapper around a wide variety of pre-trained open source neural networks of the type called Transformers [5]. They also had, on paper, some success with task C, however this was entirely due to weaknesses in the assessment mechanisms given by SemEval, for reasons we will examine later. In this paper we will explain some background on transformer technology, we explain the work that was already done on both the first two and the latter tasks, and we will explain our attempts to get improved performance, particularly at task C. While we succeeded in some ways in improving the results of the generation portion of the task, we struggled to increase the actual score we received from the SemEval scoring system. This is likely due to both the difficulty of the task itself and the aforementioned problems with the scoring system.

1. Problem Statement

1.1 Need:

The overarching goal of the system is to differentiate natural language statements that make sense from those that do not make sense. This can be achieved by completing three different subtasks: commonsense validation, explanation, and reason generating, which are denoted by subtasks A, B, and C, respectively.

Subtask A (Validation): The system is required to decide which natural language statement does not make sense when given two sentences with similar wordings.

Subtask B (Explanation): When given three options that explain why the statement in subtask A does not make sense, the system must select the most corresponding option.

Subtask C (Reason generating): The system is required to generate three more referential reasons, in the form of a sequence of words, as to why the statement in subtask A does not make sense. BLEU score will be used to evaluate this subtask.

Ultimately, we aim to create an interactive webpage that allows users to enter two natural language statements and obtain the results of the three subtasks mentioned above.

1.2 Objectives:

1. To achieve a higher BLEU score of 6.17 in Subtask C (generation of reason).
2. Create an interactive webpage that incorporates the functions listed in subtask A, B, and C.

1.3 Glossary:

BLEU((Billingual Language Understudy) Score: an algorithm used to originally to compare machine-translations against human translated sentences but used in Task C in the ComVe Challenge to compare a generated explanation against reference explanations of why a statement is nonsensical. A score closer to one meaning higher similarity to the human created reference texts

Commonsense Reasoning: a branch of AI that focuses on the programming of computers to be able to make sense of and utilize day to day human assumptions.

Commonsense Validation and Explanation(ComVe) Challenge 2020: A competition based on whether a system can evaluate whether text makes sense via three tasks. Task A is validation, deciding whether a statement is sensical. Task B is explanation (multichoice), choosing which explanation appropriately describes why a sentence is against common sense. Task C is explanation (generation), creating a sentence that explains why a sentence is against common sense and comparing it to referential explanations via BLEU.

Generative Predictive Transformer 2 (GPT-2): unsupervised transformer language model that is capable of generating natural sounding text.

Generative Predictive Transformer 3 (GPT-3): the newly developed successor to GPT2 that possesses two orders of magnitude higher number of parameters. It can create impressively human-like texts.

Natural Language Processing(NLP): the programming of computers to process and understand organically structured language.

Text-to-Text- Transfer Transformer (T5): a model for transfer learning technique created by Google where the input and output are always text strings.

2. Problem analysis and research

One of the very first obstacle we bumped into was the lack of understanding about transformer models and how they function in commonsense reasoning. Therefore, each member was assigned to look up on research or articles that will help us understand commonsense reasoning and its state-of-the-art technologies. After multiple meetings and discussions, our team has gained more confidence in utilizing different models, such as BERT, RoBERTa, GPT-2, and T5, to achieve a higher score in each subtask.

Our next agenda was to achieve a higher BLEU score for subtask C compared to what Sirwe's team has done, which is the primary goal of our entire project. More research and articles have been dug up and discussed about in effort to generate reasonings that make more sense in subtask C. Eventually, our team narrowed down the possible ways to achieve a higher score to four main methods:

- Experimenting with different models
- Finetuning hyperparameters of models
- Training models with custom dataset and dataset provided by SemEval
- Try different generation formats

While we developed some promising results based on subjective inspection of the output, we failed to increase performance according to the SemEval scoring mechanism (BLEU). However, we quickly realized that there are some flaws to the BLEU algorithm as a scoring mechanism for text generation tasks. It is limited in ability to actually detect coherence of answers, as it only assesses similarity to pre-determined 'correct' answers. Not only do good results not always score well, bad results can score better. That is, by simply feeding the scorer back to the algorithm, the prompt yields a score of 6.17, which is significantly higher than what we achieved. Nevertheless, since the SemEval officials have decided to use BLEU as their scoring benchmark, our only option is to abide by it.

3. Requirements

The interactive website should provide a text input bar that will allow users to input any sentences. The system must support the following inputs:

a) Any English words
b) Any English sentences

The system will generate an error message and request a new input from the user when given prompts such as:

a) Any non-English characters
b) Any non-English sentences
c) Any sentences made up of only numbers or symbols

If no error message is generated, the system will proceed to execute the three subtasks mentioned in *Section 1.1: Need* and print the appropriate results on the user's screen.

4. Standards and Constraints

4.1 Applicable standards

a) Development in Commonsense Reasoning
    - The code was all built using a python library called Huggingface Transformers. Huggingface provides methods for easy implementation of generic transformers, as well as a wide variety of wrappers for publicly available pre-trained models.
    - The code was mostly developed on Google Colab, a google cloud-based implementation of Jupyter notebooks.
    - Style guide for python, PEP 8, was used as a reference during the development of the project.

b) Web application
    - The web application was also developed with python, utilizing its lightweight micro web framework, Flask. It is classified as a microframework because it does not require particular tools or libraries.
    - Our flask application is built off from the admin Dashboard generated by the AppSeed platform in Flask Framework on top of Black Dashboard (free version), an open-source Bootstrap 4 dashboard template. The Flask codebase is provided with authentication, database, ORM and deployment scripts.
    - SQLite were used as our go to database while we used SQLAlchemy as our Database integration tool.

4.2 Constraints

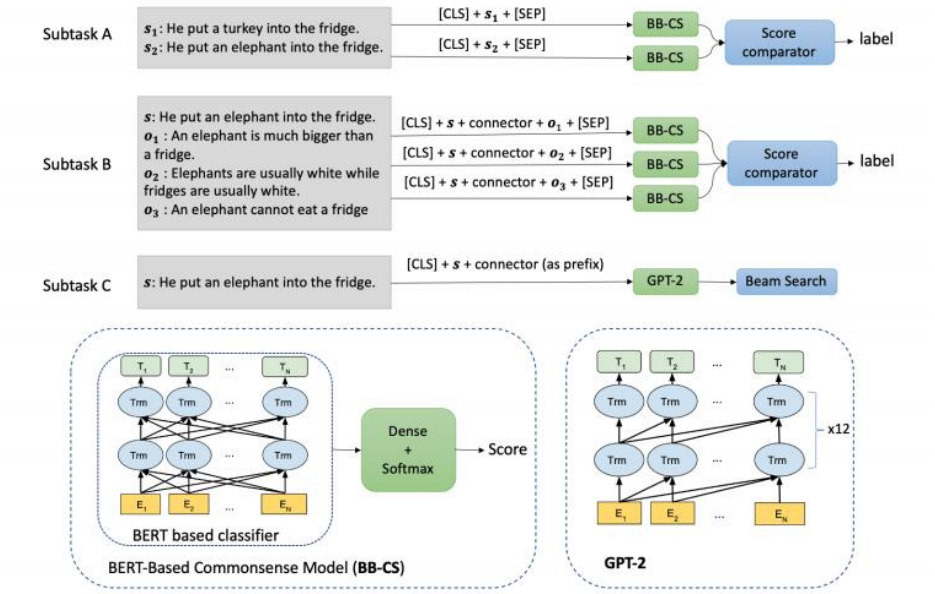
a) Timing:
   - The system and its complementary website should be available for beta-tests no later than **March 2021** to prepare for roll-out by the end of **April 2021.**

b) Reliability:
   - The system should be completely operational at least **90%** of the time.
   - Down time after a failure should not exceed **24 hours.**

c) Usability:
   - A very simple guide on how to use the system and navigate around the webpage should be provided.
   - Users should be able to fully utilize the system within **1 minute** after browsing the webpage.

d) Supportability:
   - The system website should be viewable from **Google Chrome 87.0 or later, and Mozilla Firefox 83.0 or later.**

5. System Design



Model pictured is the prototypical architecture described in the task description paper. Our system architecture has evolved and deviated from this architecture along the way. There are 5 essential components for our commonsense reasoning NLP project, and it could be visualized in the figure below:

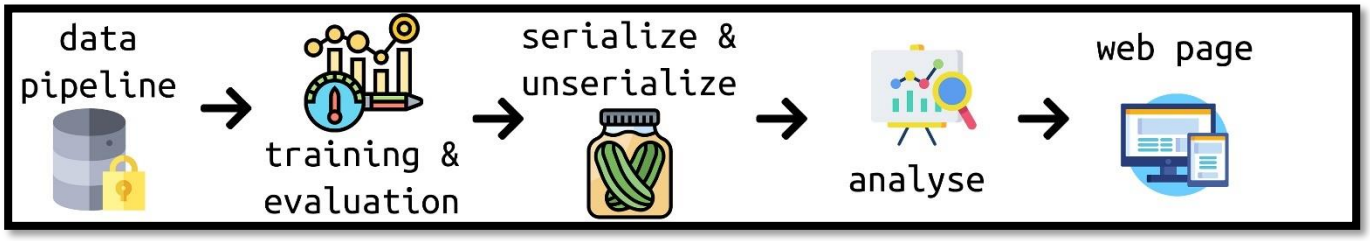


Fig 1 shows the user journey through our application, from start to finish.

a. Data Pipeline

Probably the largest component of our project, the data pipeline is an overview of how our data progress throughout the application, from data import to data preprocessing. There is no data collection required for this project as we will be using the dataset provided from the SemEval 2020 Competition available on GitHub (https://github.com/wangcunxiang/SemEval2020-Task4-Commonsense-Validation-and-Explanation/tree/master/ALL%20data). After identifying the dataset, we will then progress into data preprocessing.

i. **Data preprocessing:** In this step, we will be analyzing and modifying the given dataset given so that it would improve our model in the training step. For example, we are planning to remove punctuation at the end of the sentences when working on SubTask A as it has proven to increase result accuracy by a few percent.

b. Training and Evaluation

i. With the clean and preprocessed data, our goal is to improve the result of Subtask A, B and C. Here's we had experimented with different models such as OpenAI GPT-2, RoBERTa, BERT_Classification to determine the most effective model for each individual subtasks.
ii. We then carry out feature engineering and tuning of hyperparameters to achieve desired outcome
iii. With the data and correct hyperparameters, we trained and evaluated the trained model.

c. Serialize and deserialize model

As feature engineering and training is a heavy process in machine learning, it is important to store the final model in a reusable format.

i. Therefore, we will serialize the model once achieve desired results to preserve the accuracy of the model.
ii. We will then carry out deserialization of the model on the web application.

d. Analysis and Evaluation

Once the model is deserialize on the web application, users could directly use the trained model for evaluation.

e. Web Interface

For demonstration purposes, we have implemented a user-friendly website to demonstrate how the commonsense model work. We would then display Subtask A, B, and C on our website and allow users to interact with them. Some features include validating the commonsense of sentences inputted by the user and being able to generate an explanation as to why the selected sentence is nonsense.

Task specific specifications:
Task C specifically entailed some unique design considerations, while conforming with the above general path. The pipeline initially consisted of a prompt, such as “he put an elephant in the fridge”, concatenated with “doesn’t make sense because”, and a predictive language model was leveraged to predict the words that followed. Later the model was trained such that the “doesn’t make sense” string was replaced with a unique Seperator token, ‘<SEP>’. Without task-specific training the pre-trained models we used would not have been able to interpret this, but with our fine tuning this noticeably increased the output quality, as it established a repeating structural element for the model to generate off of. This is an excellent example of the usage of fine tuning, as the model can leverage both the advantages of being trained on a very large corpus of general purpose data, which helps it ‘know’ general sentence structure and related language constructs, while also ‘knowing’ how the specific task is organized [4].

6. Testing

Testing the model in our case was very straightforward as the primary testing architecture was built into the nature of the project. Code for testing all three of the tasks was provided by SemEval. Each of the tasks had an associated numerical score out of 100 to represent the accuracy of the results. For tasks A and B, this score represents a simple percentage of correct answers. For task C, the scoring system used an implementation of the BLEU (Bilingual Evaluation Understudy) algorithm. This is an algorithm designed for scoring machine translation quality that was adapted for this use case. It works by comparing the output sentence to 1 or more pre-determined correct outputs, (in this instance, 3) and essentially scores the result by seeing how many of the words in the generated output appear in the 'correct' outputs.

There is some debate on whether this system is ideal for scoring actual translations, and there are further problems with the prospect of adapting it to this purpose [1]. Simply feeding the scorer the inputs intended to be used to generate an output, I.e "he put an elephant in the fridge" is paired with "he put the elephant in the fridge" instead of an actual explanation generates a score of 6.17. We are aware of this because it is what the previous team used for their final submission for this project, as they were unable to produce a model that consistently got a better score, and it therefore marks our target score for task C. A primary issue with this form of translation evaluation is that there are many different possible correct translations for any given sentence, and that problem is significantly exacerbated in the instance of commonsense reasoning. For this reason it is difficult to objectively test a model like this without human grading.

Other than the SemEval testing mechanisms, we examined the results for a subjective analysis of the results.

As for the website, testing was a relatively simple matter of using the interface as intended. There is no open-ended user input on the site so there is limited possibility to create errors.

7. Results

7.1 Realization of requirements

Referring to *Section 1.2: Objectives*, we have failed to implement our first objective of the project, which is to achieve a higher BLEU score than 6.17 in subtask C, due to the difficulty in the nature of this challenge. We did, however, took a different approach in tackling subtask C, and achieve a fairly good results, regardless of the diminishing returns demonstrated by the BLEU scoring mechanism. On the other hand, we achieved success in our second main objective of the project, which is creating an interactive webpage that incorporates the functions of subtasks A, B, and C.

7.2 Realization of Standards and Constraints

Due to our inability to foresee the complication associated with subtask C, our team did delay the initially planned beta-testing date. However, the fully functional webpage was completed and rolled-out before we presented our project to the client (April 20, 2021). Besides the timing constraint, our project managed to realize the other standards and constraints.

7.3 Testing results

As previously mentioned, our SemEval scores were approximately on par with the results obtained by the previous WMU group. We failed to get an average result for task C above the 6.17 obtained by the naive input-to-output pipeline, although we did get better results than their more sincere attempts. Our website interface encountered no errors and passed the testing for each of the three task interfaces.

8. Future Work

The paths available for future work in this regard are nearly limitless. Commonsense reasoning, both on its own and as a subset of natural language processing, which itself is a subset of machine learning and artificial intelligence technology, is undergoing rapid and aggressive research from universities and corporations alike. New pretrained models are frequently released. GPT-3, the successor to GPT-2, produces extremely impressive and generalizable results, for example, and would very likely help us improve our score [3]. Unfortunately it is not available to the public currently.

Nonetheless there is a large number of models that we did not get a chance to experiment with that may prove useful for this application. Models tend to vary both on the precise architecture of the neural network and the corpus of data they are trained on. Some of these are designed with specific goals in mind, or, like for example the T5 model from Google, are designed to be able to switch between different types of tasks effectively.

There are plenty of even quite simple modifications that might yield improvements as well. As we saw simply adding in a separation token produces a noticeable increase in output quality. This reveals how the processing going on ‘under the hood’ of these models are not necessarily aligned with our intuitive thinking. Perhaps artificially simplifying the inputs and re-padding the outputs would yield better results. Other possible usage of other models could, for example, involve using a data clustering model to combine prompts of a similar structure, and then using separate models for each cluster. The only real limit is creativity and time.

9. Conclusion

Language is an incredibly complex mechanism of encoding information that is used by every human for communication. Understanding it, therefore, is a task that is equally important and difficult for software. There are many subtilties and inconsistencies that create a vast number of engineering challenges that need to be addressed in the process of doing so, and commonsense reasoning is one of them.

The ability to tell whether a statement does or does not make sense, regardless of grammatical correctness, is useful for parsing language the way it is actually used in real life. We still have a long way to go before we have completely solved this problem, and that is why organizations like SemEval create these challenges, to bring more attention to the pieces of the natural language processing puzzle that still need to be solved.

The three tasks A, B and C cover a spread, from easier to more difficult, of the kinds of challenges that we still face. Task A involves parsing, task C is a mix of parsing and reasoning, and task C is pure reasoning generation. This, the creation of novel reasoning to a problem, is what is most elusive to NLP and AI researchers.

We took advantage of the great deal of open-source work on NLP to execute our attempt at solving these problems. The highly collaborative research community in AI is a great boon to beginner entrants into the technology such as ourselves. We used the Huggingface library, which itself implements publicly available transformer models to fine tune a model for our purpose. While we did not produce stunning results on paper, the output of the models are very promising. We also created a way of visualizing what the model is doing, as much of the work being done on these topics are hidden in blog posts and research papers that take some technical expertise to understand, even though the inputs and outputs are very straightforward. This is not unlike what SemEval does in that, by making it easy for people to understand the technology, hopefully we will get more people interested in contributing to the puzzle. In this way, we think we succeeded in our unspoken goal of advancing ours and others understanding of emerging artificial intelligence technology.

10. References

[1] Tatman, Racheal. “Evaluating Text Output in NLP: BLEU at Your Own Risk”. *Towards Data Science.* January 15, 2019.

[2] Cunxiang Wang, et al. “Does It Make Sense? And Why? A Pilot Study for Sense Making and Explanation”. ACL, 2019.

[3] Brown, Tom. “Language models are few-shot learners”. OpenAI, 28 may, 2020.

[4] Hendryks, Dan et. Al. “Pretrained Transformers Improve Out-of-Distribution Robustness”. ACL, July 2020.

[5]Vaswani, Ashish, et al. “Attention is all You Need”. Google, December 2017.