SemEval-2020 Task 4: Commonsense Validation and Explanation

Mohammad Karimiabdolmaleki
University of Alberta
Edmonton, AB
karimiab@ualberta.ca

Ali Zamani
University of Alberta
Edmonton, AB
azamani1@ualberta.ca

1 Abstract

In this paper, we present machine learning models, including naive Bayes, Logistic Regression, Support Vector Machine, and pretrained language models such as Generative Pre-trained Transformer 2 (GPT2), Bidirectional Encoder Representations from Transformers (BERT) and, RoBERTa to tackle the problem of commonsense validation and explanation. For subtask A, "Distil BERT" which was fine-tuned on the dataset, outperformed other models with an accuracy of 72%, and in subtask B, RoBERTa recorded a high accuracy of 85%.

2 Introduction

With the rapid development of artificial intelligence and machine learning algorithms, computers have shown promising results at understanding context and recording better performance than humans in numerous benchmarks (Wang et al., 2017). However, commonsense validation can be a challenging task for a computer model. For instance, it takes no effort for a human to understand that *someone can put a cake into the fridge* but *she can never put an elephant into the fridge*, but it might be difficult for an artificial model to recognize the difference. Commonsense validation is one of the Natural Language Understanding (NLU) tasks that has received increasing attention throughout recent years (Wang et al., 2020b). Validation and reasoning are considered as an essential capability of a practical NLU system (Davis, 2017); consequently, it is worth attempting to develop models that can perform well on the sense-making task.

The task is to evaluate how well a model can perform in validating and explaining the commonsense of English sentences. Specifically speaking, the task comprises two subtasks (Wang et al., 2020a). In the first subtask, which is called validation, the goal is to discern which sentence makes sense and which one does not, given two sentences with similar structure and wordings. The second subtask is called explanation, in which from the three reason candidates and a statement which does not make sense, the goal is to identify the reason that is most likely to explain why the corresponding statement is against the commonsense.

3 Related Work

Commonsense validation and reasoning have been studied in various ways in natural language, from rule-based methods (Bailey et al., 2015) to transformer-based language models (Pai, 2020). Due to the emergence of high-performance machine learning models in recent years, numerous research has been trying to address the problem of commonsense validation (Wang et al., 2020a). Prior to the advent of transformer-based language models, researchers tried to tackle the problem by employing linear and nonlinear classification models (Poria et al., 2014). In this way, We decided to employ naive Bayes, Logistic Regression, and Support Vector Machine as a starting point. The reason behind this choice was the interpretability of these models compared to the complicated neural network models and the acceptable performance of the models on similar datasets. However, the result of the machine learning models on the dataset provided by SemEval-2020 task 4 (Wang et al., 2020b) suggests that binary classification machine learning models are not a good fit for the problem.

The study Pai (2020) utilized pre-trained language models and fine-tuned them for subtask A and B. The model scored a high accuracy of 95.9% for subtask A and 90.8% for subtask B, which was close to human performance on the tasks. Moreover, another study Mohammed and Abdullah (2020), participated in subtask A and applied ensembling of four different state-of-the-art

pre-trained language models (BERT (Devlin et al., 2019), ALBERT (Lan et al., 2020), RoBERTa (Liu et al., 2019), and XLNET (Yang et al., 2019)) and achieved 96.2% accuracy for subtask A. Consequently, having a pipeline of fine-tuned state-of-the-art pre-trained models is a common approach that results in superior performance. The results from the studies mentioned above encouraged us to use language models for both tasks. However, the time limit and computational resources shortage stopped us from proposing an ensemble model.

The paper (Doxolodeo and Mahendra, 2020), utilized RoBERTa for subtask A to figure out which sentence does not make sense. In subtask B, they used BERT alongside replacing the dataset with Multi Natural Language Interface (MNLI) corpus (Neelakantan et al., 2017) to have a wider range of possible explanations. The study resulted in 88.2% and 80.5% accuracy for subtasks A and B. While their approach for subtask B seems unnecessary, employing RoBERTa for a classification task was intriguing. RoBERTa is an optimized version of the Bidirectional Encoder Representations from Transformers which has tuned hyperparameters and larger mini-batches and learning rates. Thus, we aimed to use BERT language model for the validation task and RoBERTa for the reasoning task. (Ben Rim and Okazaki, 2020) have also used BERT language model to identify whether a sentence contradicts the commonsense and justify the main reason of this choice based on the data of the second task. The model obtained an accuracy of 88.7% and 85.3% for subtask A and B, respectively. On the other hand, we also followed the work in (Collins et al., 2020) which uses the pre-trained GPT Head Model (Radford and Narasimhan, 2018) to calculate perplexity scores based on the sequence of tokens in input sentences and achieves an accuracy of 75% for subtask A.

4 Methods

4.1 Data Preprocessing

In subtask A, each dataset record consists of two sentences with a label corresponding to the index of the against commonsense sentence (0 or 1). As shown in Table 1, a dataset record from subtask A is listed with its associated label. It is evident that among sentences, "He drinks apple" and "He drinks milk," "He drinks apple" is against commonsense, which is indicated by the label 0. For our binary classification purposes, using machine learning models, we separate these two sentences and assign label 1 to the sentence that makes sense and 0 to the sentence that does not make sense. The next step is applying several preprocessing techniques to the sentences (e.g., lower case conversion, removing punctuation, stopword removal, and stemming) to decrease the amount of noise and vocabulary size and to prepare the data for the binary classification task.

sentence 1	**sentence 2**	**label**
He drinks apple.	He drinks milk.	0

Table 1: An example of subtask A's data

In subtask B, each dataset record consists of four sentences with a label associated with it. The first sentence is a false sentence, and there are three possible explanations; one of them is the best explanation for the false sentence specified by the label. To be more specific, consider an example shown in Table 2, "He poured orange juice on his cereal." This is a false sentence, and three possible explanations are listed, explaining why the false sentence does not make sense. Since the label for this record is "B", thus, "Option B" is a correct explanation for the false sentence.

False sentence	**He poured orange juice on his cereal.**
Option A	Orange juice is usually bright orange.
Option B	Orange juice doesn't taste good on cereal.
Option C	Orange juice is sticky if you spill it on the table.
Label	B

Table 2: An example of subtask B's data

To prepare Task B's dataset, we first remove the end sentence punctuation at the end of sentences (i.e., 'dot') and concatenate the false sentence with each one of the option sentences to have three separate sentences. As an instance, for the record shown in Table 2, we will have the following sentences after the above-mentioned procedure:

- He poured orange juice on his cereal Orange juice is usually bright orange
- He poured orange juice on his cereal Orange juice doesn't taste good on cereal
- He poured orange juice on his cereal Orange juice is sticky if you spill it on the table

To prepare the sentences for the language modeling tasks in subtask A and B, we applied a preprocessing pipeline comprising: lower case conversion,

truncation, adding "CLS" and "SEP" tokens to the beginning and end of the sentences, tokenization, id conversions, and padding to the input sentences.

4.2 Subtask A: Validation

4.2.1 Machine Learning Models

As a baseline, we implemented naive Bayes, logistic regression, and Support Vector Machine in scikit-learn with Term Frequency-Inverse Document Frequency (TF-IDF) (Das and Chakraborty, 2018) input representations to have a taste of the data complexity. In the following, we describe implemented machine learning models.

- **naive Bayes**: Naive Bayes is a statistical classifier that is based on Bayes' theorem which assumes all features are independent. Naive Bayes uses prior probabilities and conditional probabilities to assign a label to the input records.
- **Logistic Regression**: Logistic Regression is another statistical model that is based on the sigmoid function which is suitable for binary classification tasks.
- **Support Vector Machine (SVM)**: SVM is another robust machine learning algorithm that maps the training data to points in space and tries to classify the points using a hyperplane in a way that the distance between the closest points of the two classes is maximized.

As commonsense validation is a complex problem in nature and requires comprehension and contextual awareness, we do not expect the machine learning models to perform well. Thus, we have employed several powerful language models to tackle the problem.

4.2.2 GPT-2 and BERT Language Models

Language Modelling is using probabilistic and statistical methods to calculate the probability of a given sentence or sequence of words. Pretrained language models are trained on large corpora and usually have millions of parameters. Thus, we believe that commonsense is in the nature of the pretrained language models. That being said, the language model should assign a lower probability to a sentence which does not make sense and a higher probability to a sentence which makes sense.

Following the work in (Collins et al., 2020) we employed pre-trained language models (GPT2 (Radford et al., 2019) and BERT) to estimate the probability of a sentence. We utilized 'distilgpt2', the smallest version of the GPT-2 with six layers and more than 82 million parameters, trained on OpenWebTextCorpus (Gokaslan and Cohen) and 'bert-base,' the smallest version of BERT with 12 layers and 110 million parameters, trained on BooksCorpus (Zhu et al., 2015) and English Wikipedia. The sentence is given as input to the pre-trained language model, and the output is the perplexity of the corresponding sequence. The sentence which obtains a lower perplexity is the sentence that relatively makes sense. We also fine-tuned the 'bert-base' language model on the dataset to find the best parameters and achieve robust performance. We fine-tuned the model using AdamW optimizer from Pytorch (Paszke et al., 2019) with a learning rate of 5e-5 and batch size of 32 for two epochs on 90% of the training dataset and testing the validation performance on the 10% held-out set. We have also employed 5-fold cross-validation to ensure that the prediction results are robust and accurately represent the model's power.

4.3 Subtask B: Resoning

To solve the reasoning problem, we used the RoBERTa language model, a transformers model based on BERT architecture, pretrained on a large corpus of English data. RoBERTa can train on the raw text only, without human labeling them so that it can use lots of publicly available data for training. More specifically, RoBERTa is Masked Language Model (MLM) objective. Given a sentence, it randomly masks 15% of the words in the sentence and feeds the masked sentence to the model, and the model is in charge of predicting masked words. This feature differentiates RoBERTa from autoregressive models like GPT and allows the model to learn a bidirectional representation of the sentence. RoBERTa has been trained on the five datasets: BookCorpus, English Wikipedia, CC-News (Mackenzie et al., 2020), OPenWebText (Gokaslan and Cohen), and Stories (Akoury et al., 2020). we employed "roberta-base" with 12 layers and 125 million parameters (Liu, 2020). We feed the three generated sentences mentioned in the 4.1 to the RoBERTa model for training and testing purposes.

5 Results

5.1 Subtask A: Validation

The result of the machine learning and pretrained language models are shown in Table 3. As expected, the pretrained language models outperformed other methods with scoring a higher accuracy, implying that the task requires models that can capture context.

Model	Accuracy	Precision	Recall	F1-Score
naïve Bayes	0.37	0.37	0.37	0.37
Logistic Regression	0.38	0.38	0.38	0.38
SVM	0.39	0.38	0.39	0.37
Distill GPT-2	0.68	0.68	0.67	0.67
BERT base	0.55	0.56	0.56	0.56
Tuned BERT	0.72	0.73	0.72	0.72

Table 3: Subtask A models' performance on the test data

5.2 Subtask B: Reasoning

We employed the "roberta-base" model for solving subtask B with 12 layers and 125 million parameters. The batch size was set to 32, and the learning rate is equal to 2e-5. Moreover, epsilon which is a term added to the denominator of AdamW optimizer (Kingma and Ba, 2017) to improve numerical stability, was specified as 1e-8. Since training the model required a significant amount of time (about 5 hours for each epoch), we only trained the model for one epoch and ensured that the training loss was decreasing during the training process. The model obtained an accuracy of **0.85** on the test data.

6 Discussion

The findings of this study are aligned with what was previously found by the literature in commonsense validation and explanations. Understanding the sense-making of a sentence requires a deep comprehension of the sentence.

By looking at the error logs of task A, we found several cases where there was an ambiguity between two phrases. For instance, one of the records of the dataset contains: "The chef put extra lemons on the pizza." and "The chef put extra mushrooms on the pizza.". Based on the dataset, the first sentence is considered against commonsense. However, "GPT2" predicted the second sentence as a sentence that is against commonsense. In fact, the "Grilled Lemon Pizzas" is a pizza having lemon in its ingredients, so the "The chef put extra lemons on the pizza" is not totally against commonsense. On the other hand, classifiers misclassified several instances where there was no ambiguity in the sentence. As an instance, consider the pair "He put a pig into the pan." and "He put the steak into the pan." It is obvious that the first sentence is against commonsense, while the model specified the second sentence as a sentence that is against commonsense. This phenomenon implies the fact that we had a limited time frame, and we did not have enough computational resources to fine-tune the model on the training data. We were only able to fine-tune "bert base" on the training data for subtask A which significantly increased the model's performance by around 20%. We have seen the same pattern in training RoBERTa on the subtask B's data. As a result, fine-tuning RoBERTa for a higher number of epochs would be a direction for our future work.

7 Conclusion

In most NLU systems, the ability of commonsense validation and explanation is crucial and directly affects the rationality of the generated model output (Pai, 2020). Moreover, since most of the commonsense expressions are mysterious, understanding the meaning of vocabulary is essential for classifying the data correctly, which future increase the complexity of the problem. Hence, most of state-of-the-art existing work are based on language models. Our analysis reveals that even employing machine learning models with techniques found a couple of years ago can not solve the problem. Thus, it is essential to use models that are aware of context, especially transformer-based models that use attention mechanisms and bi-directional models that can capture more information from context from two sides of the sentence. As a direction for future work if sufficient computation resources are available, one can fine-tune the transformer-based models on the training data to optimize the parameters of the model for the corresponding task. Another direction for achieving a better performance is to use ensemble models; using a combination of pretrained language models and passing the weighted sum of the output to the prediction layer .

8 Repository URL

https://github.com/UOFA-INTRO-NLP-F21/f2021-proj-zamaniali1995/

References

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. Storium: A dataset and evaluation platform for machine-in-the-loop story generation.

Daniel Bailey, Amelia J Harrison, Yuliya Lierler, Vladimir Lifschitz, and Julian Michael. 2015. The winograd schema challenge and reasoning about correlation. In *2015 AAAI Spring Symposium Series*.

Wiem Ben Rim and Naoaki Okazaki. 2020. SWAGex at SemEval-2020 task 4: Commonsense explanation as next event prediction. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 422–429, Barcelona (online). International Committee for Computational Linguistics.

Kris Collins, Max Grathwohl, and Heba Ahmed. 2020. Mxgra at SemEval-2020 task 4: Common sense making with next token prediction. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 569–573, Barcelona (online). International Committee for Computational Linguistics.

Bijoyan Das and Sarit Chakraborty. 2018. An improved text sentiment classification model using TF-IDF and next word negation. *CoRR*, abs/1806.06407.

Ernest Davis. 2017. Logical formalizations of commonsense reasoning: a survey. *Journal of Artificial Intelligence Research*, 59:651–723.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Kerenza Doxolodeo and Rahmad Mahendra. 2020. UI at SemEval-2020 task 4: Commonsense validation and explanation by exploiting contradiction. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 614–619, Barcelona (online). International Committee for Computational Linguistics.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Pai Liu. 2020. Qiaoning at semeval-2020 task 4: Commonsense validation and explanation system based on ensemble of language model.

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.

J. Mackenzie, R. Benham, M. Petri, J. R. Trippas, J. S. Culpepper, and A. Moffat. 2020. Cc-news-en: A large english news corpus. In *Proc. CIKM*, pages 3077–3084.

Roweida Mohammed and Malak Abdullah. 2020. TeamJUST at SemEval-2020 task 4: Commonsense validation and explanation using ensembling techniques. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 594–600, Barcelona (online). International Committee for Computational Linguistics.

Arvind Neelakantan, Quoc V. Le, Martin Abadi, Andrew McCallum, and Dario Amodei. 2017. Learning a natural language interface with neural programmer.

Liu Pai. 2020. QiaoNing at SemEval-2020 task 4: Commonsense validation and explanation system based on ensemble of language model. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 415–421, Barcelona (online). International Committee for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Soujanya Poria, Alexander Gelbukh, Erik Cambria, Amir Hussain, and Guang-Bin Huang. 2014. Emosenticspace: A novel framework for affective common-sense reasoning. *Knowledge-Based Systems*, 69:108–123.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Y. Zhang. 2020a. Semeval-2020 task 4: Commonsense validation and explanation. In *SEMEVAL*.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2020b. Does it make sense? and why? a pilot study for sense making and explanation.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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 *The IEEE International Conference on Computer Vision (ICCV)*.