JUSTers at SemEval-2020 Task 4: Evaluating Transformer Models Against Commonsense Validation and Explanation

Ali Fadel
Jordan University of Science & Tech
Irbid, Jordan
aliosm1997@gmail.com

Mahmoud Al-Ayyoub
Jordan University of Science & Tech
Irbid, Jordan
maalshbool@just.edu.jo

Erik Cambria
Nanyang Technological University, Singapore
cambria@ntu.edu.sg

Abstract

In this paper, we describe our team's (JUSTers) effort in the Commonsense Validation and Explanation (ComVE) task, which is part of SemEval2020. We evaluate five pre-trained Transformer-based language models with various sizes against the three proposed subtasks. For the first two subtasks, the best accuracy levels achieved by our models are 92.90% and 92.30%, respectively, placing our team in the 12^{th} and 9^{th} places, respectively. As for the last subtask, our models reach 16.10 BLEU score and 1.94 human evaluation score placing our team in the 5^{th} and 3^{rd} places according to these two metrics, respectively. The latter is only 0.16 away from the 1^{st} place human evaluation score.

1 Introduction

Addressing the issue of commonsense understanding using deep learning algorithms and models has attracted increasing attention from the research community. Building a natural language processing (NLP) system that can understand both explicit and implicit knowledge, validate it and correct it before extracting important information is a task that is both hard and complex. At the same time, this task is essential to a large number of tasks like language modeling, word sense disambiguation and sentiment analysis (Cambria et al., 2009).

Approaches like KnowBERT (Peters et al., 2019) leverage knowledge bases like WordNet (Miller et al., 1990) and DBPedia. Such knowledge bases provide a rich source of high quality, human-curated knowledge to fine-tune the internal hidden states of large language models like BERT (Devlin et al., 2018). Other approaches leverage external knowledge to enhance classification, e.g., Sentic LSTM (Ma et al., 2018) tackled the problem of targeted aspect-based sentiment analysis by embedding the knowledge from SenticNet (Cambria et al., 2020) into an attentive long short-term memory (LSTM) network.

The Commonsense Validation and Explanation (ComVE) task (Wang et al., 2020) at SemEval2020 was proposed based on a dataset built by Wang et al. (Wang et al., 2019) to evaluate deep learning algorithms and models against commonsense tasks. The general purpose of the dataset is to test whether an NLP system can differentiate statements that make sense from those that do not. It consists of three subtasks:

- **Task A: Validation** (Sentence Classification) In this subtask, the system is given two natural language statements with similar wordings and it is expected to distinguish the statement that makes sense from the one that does not. For example:

  Statement 1: He put a turkey into the fridge. *correct*
  Statement 2: He put an elephant into the fridge.

- **Task B: Explanation** (Multiple Choice) In this subtask, a natural language statement that contradicts commonsense is given along with three reasons why it does not make sense. The system is expected to choose the correct justification. For example:

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 535–542
Barcelona, Spain (Online), December 12, 2020.

Statement: He put an elephant into the fridge.
Options:

- A: An elephant is much bigger than a fridge. *correct*
- B: Elephants are usually white while fridges are usually white.
- C: An elephant cannot eat a fridge.

- **Task C: Explanation** (Text Generation) This subtask is similar to the second one since the input to both is a single natural language statement that contradicts commonsense and the goal is to determine why is that. However, in this subtask, the system is expected to generate the justification from scratch. The generated text is evaluated against three correct reference justifications. For example:

  Statement: He put an elephant into the fridge.
  Referential Reasons:
  - An elephant is much bigger than a fridge.
  - A fridge is much smaller than an elephant.
  - Most of the fridges aren't large enough to contain an elephant.

The evaluation metric for Subtasks A and B is accuracy. As for Subtask C, the evaluation involves the BLEU score as an automatic evaluation metric in addition to a human evaluation score. For more detailed discussion and analysis of the dataset, please refer to (Wang et al., 2019).

In this work, we evaluate and fine-tune five pre-trained Transformer-based (Vaswani et al., 2017) language models with various sizes for the previously mentioned commonsense subtasks. For the first subtask, we use BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2019) language models. The best we achieve is 92.90% placing our team in the 12^{th} place among 39 teams. While in the second subtask, we utilize XLNet (Yang et al., 2019) in addition to BERT and RoBERTa language models to achieve 92.30% accuracy, which places our team in the 9^{th} place out of 27 teams. Finally, in the last subtask, we used GPT-2 (Radford et al., 2019) language model to tackle the text generation task. Our system achieves 16.10 BLEU score and 1.94 human evaluation score, which places our team in the 5^{th} place (out of 17 teams) and the 3^{rd} place, respectively. Our system's human evaluation score is only 0.16 away from the top score in the competition. Our code and experimental results are publicly available in a GitHub repository.[1]

The rest of this paper is organized as follows. The following section demonstrates how we utilize the pre-trained Transformer-based language models to build our systems for each subtask. In Section 3, we present our experimental results and discuss some insights. Finally, we provide concluding remarks in Section 4.

2 System Overview

In the following subsections we describe how we utilize the pre-trained Transformer-based language models to build our subtasks' systems.

2.1 Task A: Validation (Sentence Classification)

In this subtask, we evaluate three Transformer-based language models, namely: BERT, RoBERTa and ALBERT. All of these models were built on top of BERT with different training procedures and datasets and architectural enhancements in order to improve the resulting models.

We treat this task as a binary classification problem. I.e., given two statements, one conforms with commonsense and the other against it, the correct ones are labeled with 1s and the rest with 0s. Based on this dataset, we fine-tune the language models to perform binary classification. To produce an inference using this model, we input the correct and the wrong statements and get their probabilities of being correct or not from the model independently. After that, the statement with the lower probability is considered as the one that is against commonsense. Figure 1a shows the task's training and inference procedures.

[1] https://github.com/AliOsm/SemEval2020-Task4-ComVE

2.2 Task B: Explanation (Multiple Choice)

We evaluate XLNet in addition to BERT and RoBERTa language models against the multiple choice task. The approach is straightforward. Each one of the three given options (reasons) is concatenated independently with the given statement (that is against commonsense). The concatenation is entered to the model, and the model is fine-tuned to select one of the three options as a multi-class classification problem.

In the inference phase, the same procedure in training is used to predict the correct reason based on the probabilities given by the model using a Softmax layer. Figure 1b shows the task's training and inference procedures.

2.3 Task C: Explanation (Text Generation)

For text generation task, we utilize GPT-2 language model to tackle the problem. The idea is to fine-tune the model to generate a reason that clarifies why the given statement is incorrect. The model is trained to generate the next word given the previous sequence of words. To train it, given the task dataset, we concatenate the wrong statement with each of the given referential reasons independently (with a separator in-between). Thus, from each example, we construct three training instances.

To generate a reason why the given statement is against commonsense, we input the wrong statement into the model followed by the same separator used in the training phase and ask the model to generate text token-by-token until reaching the end of text token. Figure 1c shows the task's training and inference procedures.

3 Experimental Results

To run our experiments with the targeted pre-trained Transformer-based language models, we use the Google Colab platform (Carneiro et al., 2018) and two open source Python packages, which are Transformers (Wolf et al., 2019)[2] and SimpleTransformers.[3] For each subtask, we report the development and test sets results, training time and model size. More experimental results can be found in our GitHub repository.

In the following subsections, we describe the models' types and sizes that are used for each subtask, their results and some insights learned from our experiments.

3.1 Task A: Validation (Sentence Classification)

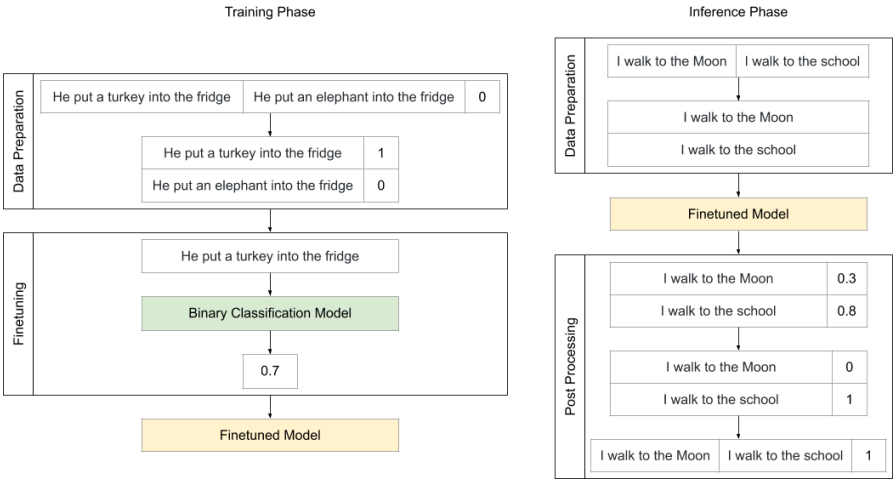
For this subtask, we use Nvidia Tesla K80 GPU from Google Colab platform to perform the experiments reported in Table 1. The learning rate is set to 4×10^{-5} and the maximum sequence length to 30 tokens in all experiments, while using training and evaluation batch sizes of 32. The models in Experiments A1 and A2 are trained for ten epochs, while the models in Experiments A3-A7 and A9 are trained for 15 epochs. Finally, Experiment A8's model is trained for 20 epochs.

As shown in the table, using larger model size consistently leads to better results as expected. A comparison between BERT's cased and uncased models (Experiments A1-A6) shows that cased models outperforms uncased models significantly, which means that the problem at hand is case-sensitive. RoBERTa base model (Experiment A7) outperforms all BERT models (base and large versions) on the development and test sets except for `bert-large-cased` model (Experiment A3), which is behind it by only 0.1%. RoBERTa large model (Experiment A8) is the best model in terms of accuracy on the test set. These results imply that RoBERTa models are more suitable than BERT models if we want to treat each of them as a knowledge base and use it to extract or validate facts. Finally, ALBERT Xxlarge model (Experiment A9) results are not as good as RoBERTa models, however, it does help when performing the ensemble models.

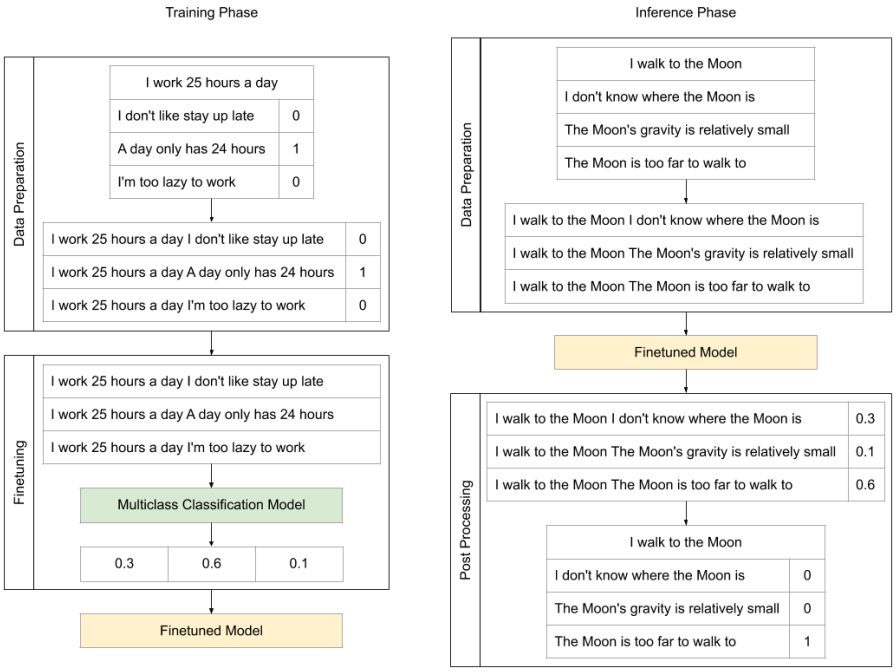
Our submission (Experiment A11) is based on majority voting ensemble between four models (Experiments A3, A7, A8 and A9). Note that we use ALBERT Xxlarge model results twice in the ensemble

[2] `https://github.com/huggingface/transformers`

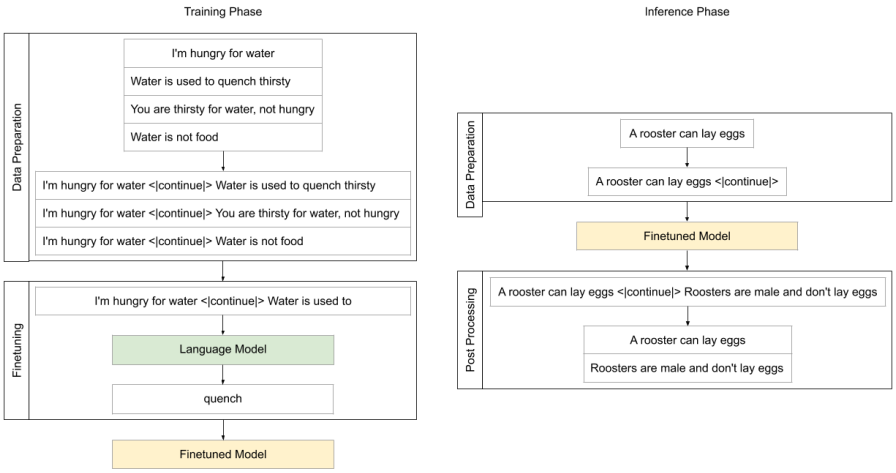
[3] `https://github.com/ThilinaRajapakse/simpletransformers`



(a) Task A



(b) Task B



(c) Task C

Figure 1: Subtasks training and inference procedures.

process, because it catches more challenging examples and improves the ensemble results on the development set. This ensemble performs better than any other model on the development set. However, RoBERTa large model outperforms it on the test set by 0.1%.

Other attempts we made to obtain better results include experimenting with different machine learning techniques, such as random forests (Breiman, 2001) with term frequency-inverse document frequency (TF-IDF) features, fastText classification (Joulin et al., 2016) and Universal Sentence Encoder (Cer et al., 2018) representations with either random forests or a simple feed-forward neural network. Among these, the best technique is using Universal Sentence Encoder representations with random forests to do the classification. This leads to 80% accuracy on the test set, which is comparable to the results of the BERT base uncased model (Experiment A2). Finally, it is worth mentioning that we also try to use paired sentences classification (Devlin et al., 2018) instead of sentence classification. However, the results do not improve.

Expr #	Model	Dev Acc.	Test Acc.	Train Time	Size
A1	bert-base-cased	85.56%	87.30%	43.64m	433.3 MB
A2	bert-base-uncased	81.54%	81.40%	44.98m	438 MB
A3	bert-large-cased	89.87%	89.40%	202.92m	1.3 GB
A4	bert-large-uncased	85.36%	83.30%	213.02m	1.3 GB
A5	bert-large-cased-whole-word-masking	88.97%	86.70%	208.60m	1.3 GB
A6	bert-large-uncased-whole-word-masking	83.05%	83.30%	199.93m	1.3 GB
A7	roberta-base	89.77%	89.30%	67.03m	501 MB
A8	roberta-large	92.78%	**93.00%**	280.51m	1.4 GB
A9	albert-xxlarge-v2	86.96%	86.60%	1050m	891.2 MB
A10	ensemble 3+6+7+8+9	92.98%	92.80%	N/A	N/A
A11	ensemble 3+7+8+9+9	**93.08%**	92.90%	N/A	N/A

Table 1: Task A: Validation (Sentence Classification) Experimental Results

3.2 Task B: Explanation (Multiple Choice)

All experiments related to this subtask (Table 2) are run on Nvidia Tesla P100 GPU from Google Colab platform. The learning rate is set to 4×10^{-5} for all experiments except for Experiments B8 and B10, where we use 1×10^{-5} as the learning rate. The maximum sequence length is set to 40 for all experiments, while using training and evaluation batch sizes of 32. The models in Experiments B1, B2, B7 and B9 are trained for five epochs, while models in Experiments B3-B6 are trained for ten epochs. Finally, models in Experiments B8 and B10 are trained for 20 epochs.

Consistent with the observation from Task A, large models achieve better results than the base versions. However, in contrast with Task A, the cased and uncased versions of BERT models (Experiments B1-B6) almost converge to the same results. Comparing RoBERTa base model (Experiment B7) results with BERT models results shows that it is on par with or better than all BERT models (base and large versions), while RoBERTa large model (Experiment B8) performs better than the other models on both the development and test sets. Hence, our submission is based on its predictions. This is consistent with our observations from Task A results. Finally, the XLNet models (Experiments B9 and B10) results are not as good as RoBERTa models, but it outperforms BERT models by significant margin on the development and test sets.

3.3 Task C: Explanation (Text Generation)

Similarly with Task B, we use Nvidia Tesla P100 GPU from Google Colab platform to run the experiments of Task C whose results are reported in Table 4. We use 5×10^{-5} as the learning rate and set the maximum sequence length to 128. We use training batch size of 64, while predicting reasons for each example independently (one example at a time). The model in Experiment C1 is trained for 15 epochs, while the

Expr #	Model	Dev Acc.	Test Acc.	Train Time	Size
B1	bert-base-cased	81.95%	80.10%	10.5m	433.5 MB
B2	bert-base-uncased	82.65%	81.50%	10.5m	438.2 MB
B3	bert-large-cased	85.46%	85.80%	76m	1.3 GB
B4	bert-large-uncased	85.46%	85.30%	74.5m	1.3 GB
B5	bert-large-cased-whole-word-masking	87.26%	87.10%	74.5m	1.3 GB
B6	bert-large-uncased-whole-word-masking	88.16%	88.30%	74.5m	1.3 GB
B7	roberta-base	86.86%	87.50%	11.16m	500 MB
B8	roberta-large	**92.38%**	**92.30%**	149m	1.4 GB
B9	xlnet-base-cased	84.25%	85.40%	10m	470.1 MB
B10	xlnet-large-cased	90.27%	88.70%	177m	1.4 GB

Table 2: Task B: Explanation (Multiple Choice) Experimental Results

models in Experiments C2 and C3 are trained for five epochs. Finally, the model in Experiment C4 is trained for one epoch only.

Using a small model, like distilled GPT-2 (Experiment C1) from HuggingFace (Sanh et al., 2019),[4] gives good results compared with the larger versions of GPT-2 models. We notice that better BLEU scores are achieved as we use larger models. This trend changes when we reach the GPT-2 large model (Experiment C4), which produces lower BLEU score on development and test sets compared with GPT-2 medium model (Experiment C3). This decrease may have been caused by the huge amount of parameters in GPT-2 large model ($\sim$774M) and the small size of the dataset. Even fine-tuning the model for less than one entire epoch did not lead to better results. We do not use GPT-2 xlarge model because of memory issues on a single 16GB GPU. Our submission is based on GPT-2 medium model predictions.

Table 3 shows some examples from development dataset paired with their generated reasons. In the first five examples, there is a trend where the model negates the entered sentence without producing a factual reason. We can see that clearly in the fifth example, where the model negates the two parts of the statement even if the second part "and can roll" is correct and this trend appears frequently in the generated reasons. The bias for negating the entered sentence that appears in the generated reasons could be attributed to a bias in the training dataset. On the other hand, for the last five examples, the model generates very clear reasons why the entered statement is against commonsense. For example, in "you can eat mercury" statement, the model generated "mercury is poisonous". So, we can safely say that the model knows what mercury is and knows some of its properties.

Against Commonsense Statement	Generated Reason
Chicken can swim in water.	Chicken can't swim.
shoes can fly	Shoes are not able to fly.
Chocolate can be used to make a coffee pot	Chocolate is not used to make coffee pots.
you can also buy tickets online with an identity card	You can't buy tickets with an identity card.
a ball is square and can roll	A ball is round and cannot roll.
You can use detergent to dye your hair.	Detergent is used to wash clothes.
you can eat mercury	mercury is poisonous
A gardener can follow a suspect	gardener is not a police officer
cars can float in the ocean just like a boat	Cars are too heavy to float in the ocean.
I am going to work so I can lose money.	Working is not a way to lose money.

Table 3: Against Commonsense Statements with Generated Reasons

It is worth mentioning that we try to do an ensemble between the best three models (Experiments C2-C4) by entering the reasons generated by GPT-2 model into the best model from Task B and using it

[4] https://huggingface.co

to select the best reason. However, the results do not improve.

Expr #	Model	Dev BLEU	Test BLEU	Train Time	Size
C1	distilgpt2	13.7582	13.8026	29m	354.2 MB
C2	gpt2	14.0547	13.6534	15m	549.5 MB
C3	gpt2-medium	**16.7153**	**16.1187**	39.33m	1.5 GB
C4	gpt2-large	16.5110	15.9299	35.66m	3.2 GB

Table 4: Task C: Explanation (Text Generation) Experimental Results

We note that the results we have discussed so far for Task C are based on the BLEU metric. Despite its overwhelming popularity in text generation tasks, this metric is known to have many flaws (Zhao et al., 2019). Fortunately, for the task at hand, the organizers provide human evaluation scores for subsets of the reasons generated by the participating systems computed as follows. They asked three human annotators to evaluate 100 randomly selected samples of the test set for each system. The rubrics are as follows.[5]

0. The reason is not grammatically correct, or not comprehensible at all, or not related to the statement at all.

1. The reason is just the negation of the statement, or a simple paraphrase. Obviously, a better explanation can be made.

2. The reason is relevant and appropriate, though it may contains a few grammatical errors or irrelevant parts. Or, it might be like case 1, but it is hard to write a proper reason.

3. The reason is appropriate and is a solid explanation of why the statement does not make sense.

Our system obtains a human evaluation score of 1.94, which means that the expected reasons outputted from it are not perfect, but they are relevant and appropriate (if the given statement is not difficult to justify why it is against commonsense). On the other hand, there are systems that achieved higher BLEU scores than our system, but their human evaluation scores are much lower than our systems. They are even close to 1, which means that the expected reasons outputted from them are not much better than a simple negation or paraphrasing of the given statements. At the end of the day, a model as simple as ours with its ability to achieve competitive results in the first two tasks and very satisfactory results for the last task, represents an appealing option for a production system for the task at hand. To further aid the reproducibility and practicality of our work, we provide our Task C fine-tuned models at HuggingFace models hub, [6] where anyone can use them using four lines of code.

4 Conclusion

In this work, we evaluated pre-trained Transformer-based language models against three commonsense tasks as part of the Commonsense Validation and Explanation (ComVE) task, which is part of SemEval2020. Our experiments showed that pre-trained language models can be treated as powerful knowledge bases to extract and validate facts. We were ranked in the 12^{th} and 9^{th} for first and second subtasks, respectively. As for third subtask, we were ranked in the 5^{th} and 3^{rd} places using automatic (BLEU) and human evaluation metrics, respectively. We provide the code and the experimental results through our GitHub repository.

Acknowledgments

We thank Google Colaboratory[7] for providing such platform that can be used to run machine learning and deep learning experiments on accelerated hardware like Nvidia Tesla GPUs for free.

[5] https://bit.ly/2W8NF0W
[6] https://huggingface.co/models?search=ComVE
[7] https://colab.research.google.com

References

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Erik Cambria, Amir Hussain, Catherine Havasi, and Chris Eckl. 2009. Common sense computing: From the society of mind to digital intuition and beyond. In Julian Fierrez, Javier Ortega, Anna Esposito, Andrzej Drygajlo, and Marcos Faundez-Zanuy, editors, *Biometric ID Management and Multimodal Communication*, volume 5707 of *Lecture Notes in Computer Science*, pages 252–259. Springer, Berlin Heidelberg.

Erik Cambria, Yang Li, Frank Z Xing, Soujanya Poria, and Kenneth Kwok. 2020. Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis. In *CIKM*.

Tiago Carneiro, Raul Victor Medeiros Da Nóbrega, Thiago Nepomuceno, Gui-Bin Bian, Victor Hugo C De Albuquerque, and Pedro Pedrosa Reboucas Filho. 2018. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6:61677–61685.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

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. *arXiv preprint arXiv:1909.11942*.

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. *arXiv preprint arXiv:1907.11692*.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In *AAAI*, pages 5876–5883.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Matthew E Peters, Mark Neumann, IV Logan, L Robert, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. *arXiv preprint arXiv:1906.00363*.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of The 14th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

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*, pages 5754–5764.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv:1909.02622*.