# TeamJUST at SemEval-2020 Task 4: Commonsense Validation and Explanation Using Ensembling Techniques

**Roweida Mohammed and Malak Abdullah**
Jordan University of Science and Technology
Irbid, Jordan
roweida.221@gmail.com
mabdullah@just.edu.jo

## Abstract

Common sense for natural language processing methods has been attracting a wide research interest, recently. Estimating automatically whether a sentence makes sense or not is considered an essential question. Task 4 in the International Workshop SemEval 2020 has provided three subtasks (A, B, and C) that challenges the participants to build systems for distinguishing the common sense statements from those that do not make sense. This paper describes TeamJUST's approach for participating in subtask A to differentiate between two sentences in English and classify them into two classes: common sense and uncommon sense statements. Our approach depends on ensembling four different state-of-the-art pre-trained models (BERT, ALBERT, Roberta, and XLNet). Our baseline model which we used only the pre-trained model of BERT has scored 89.1, while the TeamJUST model outperformed the baseline model with an accuracy score of 96.2. We have improved the results in the post-evaluation period to achieve our best result, which would rank the 4th in the competition if we had the chance to use our latest experiment.

## 1 Introduction

Natural Language Processing and Understanding (NLP/NLU) have great care of research, recently. Several language models had been trained on huge data of corpora (Peters et al., 2018; Devlin et al., 2018), and some benchmarks showed how methods display enhanced performance (Devlin et al., 2018). However, most end-to-end trained methods on common sense, compared to people, are disappointing. For instance, people can understand and recognize immediately that we can place an apple into a fridge, but cannot and never place a TV into a fridge. Thus, for trained systems, it is much harder to recognize the difference. Accordingly, it is essential to have the ability to evaluate how good a system is to recognize the sense (Davis, 2017). Recent datasets have experimented common sense over tasks, for example co-reference resolution (Ortiz, 2015) or subsequent event prediction (Zellers et al., 2018). They stated that a method is prepared with common sense through testing if it can offer a correct answer where no added knowledge to the input.

SemEval is the International Workshop on Semantic Evaluation, which is developed from SensEval to evaluate semantic study methods. The 14th version of this workshop, SemEval-2020, has provided 12 tasks. We have participated in Task 4 - Commonsense Validation and Explanation (Wang et al., 2020), which offers three subtasks with datasets in English. The main goal of this task is to recognize which sentences are common sense. Further information about Task 4 and the datasets is given in Section 3.

This paper describes our model for participating in SemEval-2020 (Task 4 - Sub Task A). Our team has built an ensembling system to select from two English statements that both have alike words to identify which one is making sense and which one is not making sense. We have used four different state-of-the-art pre-trained models (BERT (Devlin et al., 2018), ALBERT (Lan et al., 2019), Roberta (Liu et al., 2019), and XLNet (Yang et al., 2019)), then we combined their outputs and used voting method for each model to choose the shared output from each model. Our baseline model has scored 89.1 accuracies while our improved model has shown significant performance over the baseline model with scoring 96.2 and it is

*Proceedings of the 14th International Workshop on Semantic Evaluation*, pages 594–600
Barcelona, Spain (Online), December 12, 2020.

0.8 away from the first ranked model. Our baseline model ranked 17 out of 41 teams while TeamJUST would rank fourth.

The paper is constructed as follows: Related work is provided in section 2. A description for task 4 and the datasets are presented in section 3. The architecture of our approach is introduced in section 4. The detailed experiments are provided in section 5. Results and analysis are presented in Section 6. Finally, the conclusion is in section 7.

## 2 Related work

In our digital world, there is an increasing need to manage the vast amount of text. Therefore, the researchers are motivated to use the semantic knowledge to develop the models that can have a meaningful understanding of text (Liu and Singh, 2004). In (Wang et al., 2019), the researchers released a benchmark to test whether a method can distinguish the statement that is making sense, and find the reason behind why a statement does not make sense. Moreover, the researcher in (Roemmele et al., 2011) highlighted the importance of actions and consequences. They worked on finding the appropriate cause or outcome of the evidence from given two replacements. All pieces of evidence and replacements are modest sentences. For instance, 'Evidence: The girl broke her toe. What was the reason for this?' 'Replacements 1: she got in her sock a hole.', 'Replacement 2: she dropped on her foot a hammer.'

The task of Winograd Schema Challenge (WSC) (Levesque et al., 2012; Ortiz, 2015) required extra commonsense knowledge. For instance, "The crown would not fit in the black suitcase because it was too big (small). What was too big (small)?" "Answer0: the crown", "Answer1: the suitcase". Yet, they had to estimate common sense incidentally and without explaining why this answer was true whereas the other was incorrect. For example, the work in (Liu et al., 2017) used the technique of knowledge acquisition and a common model of neural association. They extracted a big number of pairs of cause-effect from different text corpus then the knowledge that was extracted was utilized to train the model of neural association.

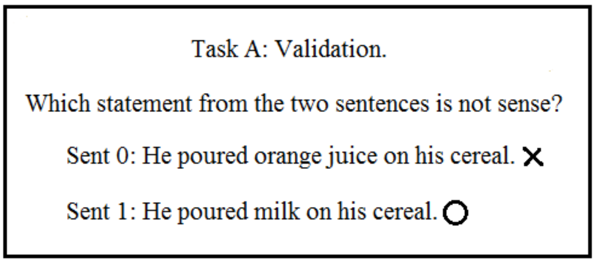
In (Mostafazadeh et al., 2016; Sharma et al., 2018) , the researchers developed a system to find out the correct finish of a story from two sentences after the fourth sentence. On the other hand, the researchers in (Ostermann et al., 2018a) provided a story text and different questions with two different answers knowing that some questions need knowledge outside the truths stated in the text. They used two models of unsupervised learning which were using only distributional information and the information of word and used two models of neural networks that were supervised.

In (Sap et al., 2019), the researchers offered a great daily commonsense rational knowledge diagram that had relations of nine if-then with variables, containing reasons and effects. They presented models of neural networks that can learn to know about previous events that are not seen to produce their possible reasons and effects in natural language. While in (Rashkin et al., 2018), the researchers suggested a new task and corpus that discover the mentioned or unmentioned people's aims and reactions beneath various everyday circumstances. They used different neural models of encoder-decoder such as (convolutional neural network and bi-directional RNN).

The goal of our system is to predict which statement is not making sense between two almost identical sentences. Our system is different than the others by using state-of-the-art pre-trained models.

## 3 Task and Data Description

Task 4 (Wang et al., 2020) in the SemEval-2020 workshop offers three subtasks with datasets in English. Subtask A is asking the users to build a system to recognize which sentences are sense from those that are not. Each record of the provided dataset contains two sentences: [sent0, sent1]. The two sentences are comparable sentences that are in similar syntax structure and only different in some words. One of the two sentences is sense while the other sentence is not. Figure 1 shows an example of the training data. For this task, the model has to detect which one is not sense.



Task A: Validation.

Which statement from the two sentences is not sense?

Sent 0: He poured orange juice on his cereal. ✗

Sent 1: He poured milk on his cereal. ○

Figure 1: Task A Validation Example.

The task's organizers have provided training, developing, trail, and testing datasets in English language. They also have chosen the accuracy score as the evaluation criteria in this competition. Table 1 shows the number of examples in each dataset. For evaluating our model and calculating the accuracy in our approach, we have used the test dataset.

| Dataset | Number of examples |
|---|---|
| Train dataset | 10,000 |
| Dev dataset | 997 |
| Trail dataset | 2021 |
| Test dataset | 1000 |

Table 1: Datasets of SemEval-2020 Task 4.

## 4 TeamJUST Model

Our model follows the workflow that is shown in Figure 2.

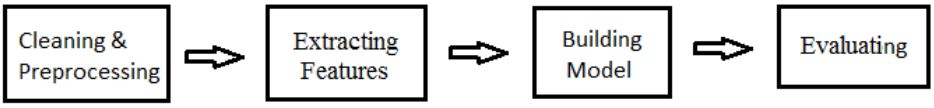


Figure 2: Workflow of the model.

### 4.1 Cleaning and Preprocessing Data

The dataset of training, trail, dev, and test have been converted to lowercase. Also, we have removed additional whitespaces and special characters. Then, we have removed the punctuation and changed the format of the dataset to csv format, with five columns and as follows: (A) *guid:* ID for the row. (B) *label:* label for the row and it should be int. (C) *alpha:* column of the same letter for all the rows and it is not used in the classification, but it is needed. (D) *text_a:* the text for the sent 0. (E) *text_b:* the text for the sent 1.

We have noticed that the target labels in the training dataset are balanced, and each label (0 stands for sent0 and 1 for sent1) has a count of 5,000. Also, we have counted the number of words in the dataset before preprocessing and after it, Table 2 shows the count of words.

| Sentences | Before Preprocessing | After Preprocessing |
|---|---|---|
| Sent0 | 77358 | 76775 |
| Sent1 | 77407 | 76830 |

Table 2: Number of words in the datatset before and after preprocessing.

From the table, we can see that the number of words in both sentences is almost equal. It is worth mentioning that we have combined training, trail, and dev dataset together to train our model.

### 4.2 Extracting Features

After preprocessing step and preparing the dataset and before training the model, we have tokenized the dataset by using the open-source framework Transformers (Wolf et al., 2019) with a maximum sequence length of 128 and then changed the text to numerical values, which can be applied to the neural network.

### 4.3 The Model Architecture

After preparing the dataset and ensuring that it is ready to be trained in our model, we fed our data into four sub-models:

**Sub-Model1:** The first sub-model trains the dataset by using the BERT model, which uses a bidirectional Transformer technique (Vaswani et al., 2017). The transformer is a mechanism of attention that learns the background relatives between words in a text.

**Sub-Model2:** This sub-model consists of the pre-trained model of ALBERT (Lan et al., 2019). The ALBERT pre-trained model offers two-parameter decrease methods to lower the consumption of memory and increase the BERT's training speed by splitting the matrix of embedding into two lesser matrices and using iterating layers divided among sets.

**Sub-Model3:** The third sub-model uses the pre-trained model of RoBERTa (Liu et al., 2019). It is defined as a Robustly Optimized BERT Pretraining that adjusts BERT's hyperparameters. It has the same design as BERT but utilizes a byte pair encoding (BPE) as a tokenizer.

**Sub-Model4:** The last sub-model is XLNet pre-trained model (Yang et al., 2019), it is an extension of the pre-trained model Transformer XL (Dai et al., 2019). It uses an autoregressive technique to learn contexts of bidirectional via maximizing the predictable probability over the changes of the order input sequence.

The predictions of the four sub-model are fed into a voting method to choose the answer. Figure 3 shows the structure for our system. We have used for each sub-model a training batch size of 20, a learning rate of 4e-05, an Adam optimizer, a dropout of 0.1, with 15 epochs.

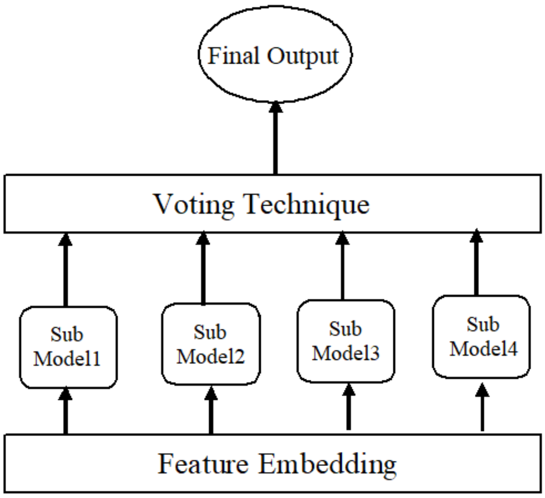


Figure 3: The structure for our system.

## 5 Experiments

We have experimented with multiple states of the art language models that have been trained on large texts. In this section, we will summarize our experiments with their results, and then we will show the result of the ensembling techniques.

**First Experiment:** In our first experiment, we have used LSTM (Hochreiter and Schmidhuber, 1997) a recurrent network architecture. We have cleaned the training dataset first and combined the two sentences and changed the output to be a binary classification. For the optimizer, we have chosen Adam (Diederik and Jimmy, 2014) and did not fine-tune the hyperparameters.

**Second Experiment:** For the next experiment, we also have applied LSTM, but with a different optimizer, RMSprop (Tieleman and Hinton, 2012) and we have fine-tuned the hyperparameters.

**Third Experiment:** For the third experiment, we have chosen a deeper recurrent neural network, BiLSTM, with RMSprop optimizer and we have fine-tuned the hyperparameters.

**Fourth Experiment:** In the fourth experiment, we have used the state of the art model BERT without fine-tuning the hyperparameters. We have also combined the sentences.

**Fifth Experiment:** For the fifth experiment, we also have used BERT, but we have fine-tuned the hyperparameters with selecting the bert-base-cased model.

**Sixth Experiment:** In the sixth experiment, we have continued using BERT, but we have used the bert-large-cased model. It is worth mentioning that after preprocessing the dataset we haven't combined the two sentences, rather we trained the model on pair sentences.

**Seventh Experiment:** In this experiment, we have used the same techniques as in the sixth experiment, but we haven't preprocessed the dataset or cleaned it.

**Eighth Experiment:** For the last experiment, which is our winning the ensembling technique, we have combined the train, dev, and trail dataset. We haven't cleaned the dataset and used it as is. We have used four different state-of-the-art pre-trained models (BERT, ALBERT, Roberta, and XLNet). Each model gives a prediction. Then, we have used the voting method to choose the mot shared answer between the four models. Figure 3 shows the structure for our system.

## 6 Results and Analysis

In Table 3, we can notice that when using ensembling techniques, we get higher accuracy compared to the other experiments we used. We can see how different state-of-the-art models have the ability to evaluate if a sentence is a common sense or not.

For the other experiments, when using different models of recurrent network architecture like LSTM or BiLSTM, it does poorly identify the sentences which are against sense, even if fine-tuning them. On the other hand, we can find a remarkable case proving that the fine-tuning process helps the model recognize common sense.

| Models | Accuracy |
|---|---|
| First Experiment | 62.0% |
| Second Experiment | 66.3% |
| Third Experiment | 68.6% |
| Fourth Experiment | 87.1% |
| Fifth Experiment | 87.9% |
| Sixth Experiment | 91.4% |
| Seventh Experiment | 91.7% |
| Eighth Experiment | 96.2% |

Table 3: Experimental Results.

## 7 Conclusion

We have investigated multiple experiments and used different models with fine-tuning them for evaluating if a system can predict if a sentence makes sense or not. The evaluating models are trained on a large raw text and we used four different state-of-the-art models. Results show how state-of-the-art models with fine-tuning them can have a significant improvement in common sense tasks.

## References

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. *Deep contextualized word representations*, arXiv preprint arXiv:1802.05365.

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

Leora Morgenstern and Charles L. Ortiz. 2015. *The winograd schema challenge: evaluating progress in commonsense reasoning*, Twenty-Seventh IAAI Conference.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. *Swag: A large-scale adversarial dataset for grounded commonsense inference*, arXiv preprint arXiv:1808.05326.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. *The winograd schema challenge*, Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. *Choice of plausible alternatives: An evaluation of commonsense causal reasoning*, 2011 AAAI Spring Symposium Series.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. *A corpus and evaluation framework for deeper understanding of commonsense stories*, arXiv preprint arXiv:1604.01696.

Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. 2018. *Tackling the story ending biases in the story cloze test*, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2:752–757.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018a. *Mcscript: A novel dataset for assessing machine comprehension using script knowledge*, arXiv preprint arXiv:1803.05223.

Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. *Atomic: An atlas of machine commonsense for if-then reasoning*, Proceedings of the AAAI Conference on Artificial Intelligence,33:3027–3035.

Hannah Rashkin, Maarten Sap, Emily Allaway, Noah A. Smith, and Yejin Choi. 2018. *Event2mind: Commonsense inference on events, intents, and reactions*, arXiv preprint arXiv:1805.06939.

Hochreiter, S. and Schmidhuber, J. 1997. *Long short-term memory*, Neural computation, 9(8), pp.1735-1780.

Diederik P. Kingma and Jimmy Lei Ba. 2014. *Adam : A method for stochastic optimization*, arXiv:1412.6980v9.

Tieleman, T. and Hinton, G. 2012. *Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude*, COURSERA: Neural networks for machine learning, 4(2), pp.26-31.

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*, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics. https://www.aclweb.org/anthology/P19-1393. 10.18653/v1/P19-1393. 4020–4026.

Hugo Liu and Push Singh . 2004. *ConceptNet—a practical commonsense reasoning tool-kit*, BT technology journal.Springer.211–226.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yi-long 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.

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.

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

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.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. *Xlnet: Generalized autoregressive pretraining for language understanding*. Advances in neural information processing systems.5754–5764.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. *Transformer-xl: Attentive language models beyond a fixed-length context*. arXiv preprint arXiv:1901.02860.

Liu Q, Jiang H, Evdokimov A, Ling ZH, Zhu X, Wei S, and Hu Y. 2017. *Cause-Effect Knowledge Acquisition and Neural Association Model for Solving A Set of Winograd Schema Problems*. In IJCAI 2017 Aug 19 (pp. 2344-2350).

Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, and Brew J. 2019. *Huggingface's transformers: State-of-the-art natural language processing.*. ArXiv, abs/1910.03771.