niksss at SemEval-2022 Task 7: Transformers for Grading the Clarifications on Instructional Texts

Nikhil Singh
Manipal University Jaipur
nikhil3198@gmail.com

Abstract

This paper describes the 9th place system description for SemEval-2022 Task 7. The goal of this shared task was to develop computational models to predict how plausible a clarification made on an instructional text is. This shared task was divided into two Subtasks A and B. We attempted to solve these using various transformers-based architecture under different regime. We initially treated this as a text2text generation problem but comparing it with our recent approach we dropped it and treated this as a text-sequence classification and regression depending on the Subtask.

1 Introduction

Instructional texts which are in the form of step-by-step instruction to achieve a particular goal are sometimes ambiguous to make out what is being talked about. To ensure that instructions describe clearly enough what steps must be followed, some clarifications are made in the places of ambiguity. This task (Roth et al., 2022) revolves around automating the grading of a particular clarification made on the instructions into plausible implausible and neutral (SubTask A) and on a finer scale where it is required to rank potential clarifications from 1 to 5 (SubTask B).

Previous work, related to this involves a Shared Task (Roth and Anthonio, 2021) which was a binary classification task, in which systems had to predict whether a given sentence in context requires clarification or not. This shared task uses the same dataset that is the wikiHowToImprove dataset (Anthonio et al., 2020) but with some variations. Instead of a binary classification task, this task is shaped as a cloze task in which, clarifications are presented as possible fillers and systems have to score how well each filler plausibly fits in a given context. A data instance can be seen in Figure 1.

Seeing the performance of BERT over BiLSTMs in (Bhat et al., 2020), we decided to build upon that and explore the use of other transformers-based models.

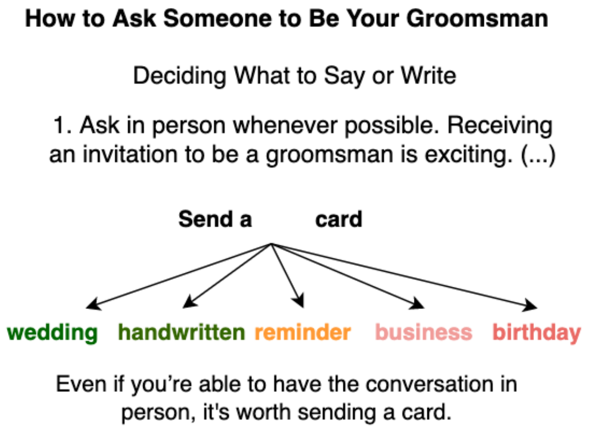

How to Ask Someone to Be Your Groomsman
Deciding What to Say or Write
1. Ask in person whenever possible. Receiving an invitation to be a groomsman is exciting. (...)
Send a card
wedding
handwritten
reminder
business
birthday
Even if you're able to have the conversation in person, it's worth sending a card.

Figure 1: A data instance of Task 7

2 System Overview

Data Pre-Processing

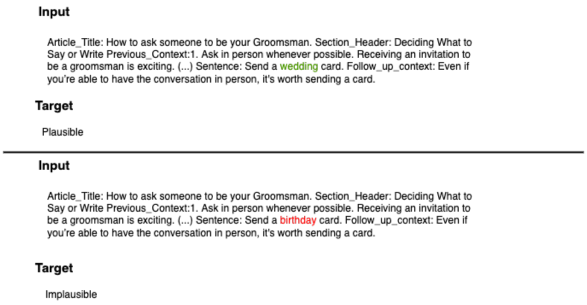
In order to convert the provided data into the form required for a text sequence classification model, we filled the blanks in the provided text with the provided potential fillers and associated their respective labels as shown in Figure 2. The resulting data were then divided into three parts training(80 %), validating(10 %), and testing(10 %).

2.1 Subtask A

For this task, we mainly experiment with two transformers-based models that differ fundamentally in the manner they were trained. Initially, we treated this as a text to text generation task using T5 (Raffel et al., 2019). The detailed steps involved in this experiment is present below.

- The Article title, previous context, the sentence (with filler), and the follow-up context was sequentially laid out one after the other.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1090 - 1093
July 14-15, 2022 ©2022 Association for Computational Linguistics



Input

Article_Title: How to ask someone to be your Groomsman. Section_Header: Deciding What to Say or Write Previous_Context:1. Ask in person whenever possible. Receiving an invitation to be a groomsman is exciting. (...) Sentence: Send a wedding card. Follow_up_context: Even if you're able to have the conversation in person, it's worth sending a card.

Target

Plausible

Input

Article_Title: How to ask someone to be your Groomsman. Section_Header: Deciding What to Say or Write Previous_Context:1. Ask in person whenever possible. Receiving an invitation to be a groomsman is exciting. (...) Sentence: Send a birthday card. Follow_up_context: Even if you're able to have the conversation in person, it's worth sending a card.

Target

Implausible

Figure 2: Data input for both T5 and BERT

- Keywords such as Article Title,Section Header,Previous Context,Sentence and Follow up context were included in the input string to indicate the respective content for the T5 model.

- The model was trained in a supervised manner using Cross-Entropy for 2 Epochs with a batch size of 2 and gradient accumulation steps of 8 making an effective batch size of 16. The rest of the Hyper-parameters were as follows:max seq length=512,learning rate=3e-4,adam epsilon=1e-8 and a seed value of 42 to keep the model deterministic.

- The model took approximately 1.5 hours to train on Nvidia's P100 GPU with a memory of 16Gb.

- The complete experiment was done on Google Colab Pro.

- The model architecture can be seen in Figure 3.

The second model used was a BERT (Devlin et al., 2018) based model.

The detailed steps involved in this experiment is present below.

- The pre-processing of the data was the same as T5 but with all the instruction constituents clubbed into a single text, i.e. all the keywords such as Article Title,Section Header,Previous Context,Sentence and Follow up context were dropped and the resulting sequence was a continuous text sequence.

- The resulting input sequence was tokenized using a BertTokenizer from Huggingface and is passed through the bert-base-uncased model to embed it into a 768 dimensional feature vector containing the syntactical information of the input string.

Model	Accuracy(%)
T5-base	40.28
bert-base-uncased	44.40

Table 1: Result on the hold-out test set for Subtask A

- The feature vector is then passed through a dropout layer to increase the regularization which in-turn increases the generalizability of the model.

- The model was trained in a supervised manner in a multiclass classification regime for 5 Epochs with a batch size of 32. Rest of the Hyper-parameters are shown in Table 2. A seed value of 42 to keep the model deterministic.

- The model took approximately 25 minutes to train on Nvidia's P100 GPU with a memory of 16Gb.

- The complete experiment was done on Google Colab Pro.

- The model architecture can be seen in Figure 4

When compared to the T5 model the BERT-based model was not just effective but also more efficient and took lower time for training and inference and therefore became our official submission for this subtask. See Table 1 for results on the hold-out test set.

The Experiment setup has been shown in Table 2. All the Experiments were performed using Huggingface Transformers Library. [1]

2.2 Subtask B

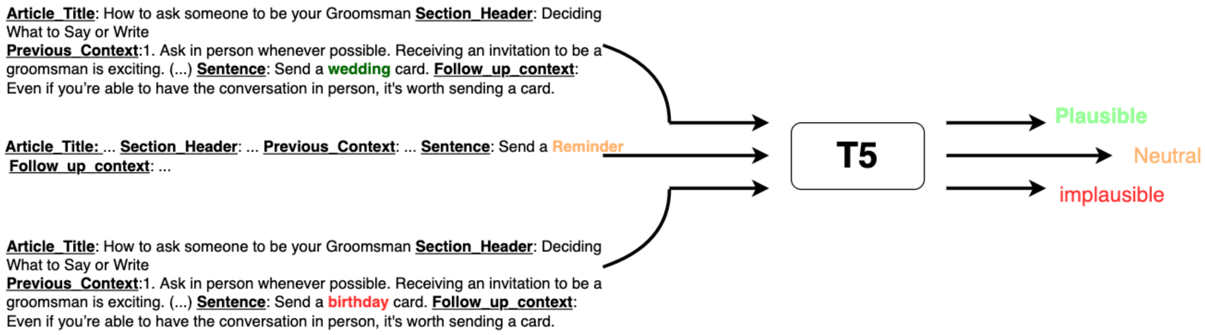
Seeing the success of BERT over T5, we make necessary changes to convert the model used for Subtask A into a regression model. See Figure 5. See Table 3 for results on the hold-out test set. The experiment setup can be seen in Table 4.

3 Results

3.1 Subtask A

The submitted systems were evaluated using the Accuracy metric(Sklearn footnote). We were officially ranked 7th in Subtask A with an accuracy

[1] https://huggingface.co/



Article_Title: How to ask someone to be your Groomsman **Section_Header**: Deciding What to Say or Write **Previous_Context**:1. Ask in person whenever possible. Receiving an invitation to be a groomsman is exciting. (...) **Sentence**: Send a **wedding** card. **Follow_up_context**: Even if you're able to have the conversation in person, it's worth sending a card.

Article_Title: ... **Section_Header**: ... **Previous_Context**: ... **Sentence**: Send a Reminder **Follow_up_context**: ...

Article_Title: How to ask someone to be your Groomsman **Section_Header**: Deciding What to Say or Write **Previous_Context**:1. Ask in person whenever possible. Receiving an invitation to be a groomsman is exciting. (...) **Sentence**: Send a **birthday** card. **Follow_up_context**: Even if you're able to have the conversation in person, it's worth sending a card.

Figure 3: Working of T5 Model

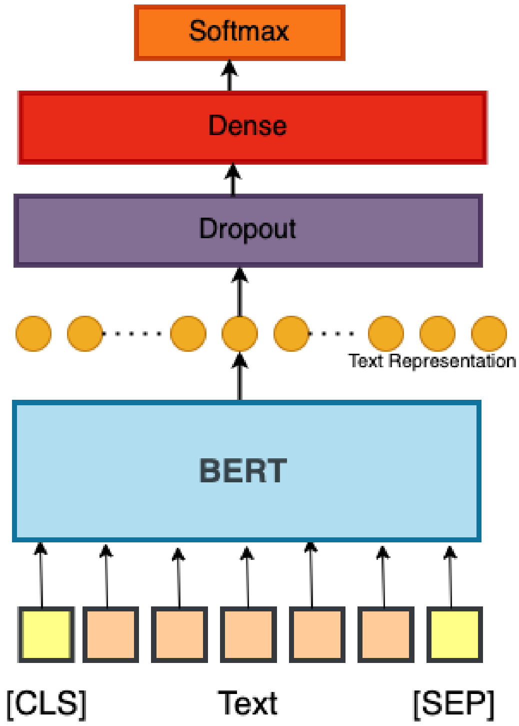


Figure 4: Model Architecture for Subtask A

Parameter	Value
Model	bert-base-uncased
Max sequence Length	256
Batch Size	8
Learning rate	2e-5
Weight decay	Linear
Momentum	0.9
Optimizer	AdamW [2]
Epochs	5
Loss	Cross Entropy

Table 2: Experimental Setup for Subtask A

Model	MSE
bert-base-uncased	44.40

Table 3: Result on the hold-out test set for Subtask B

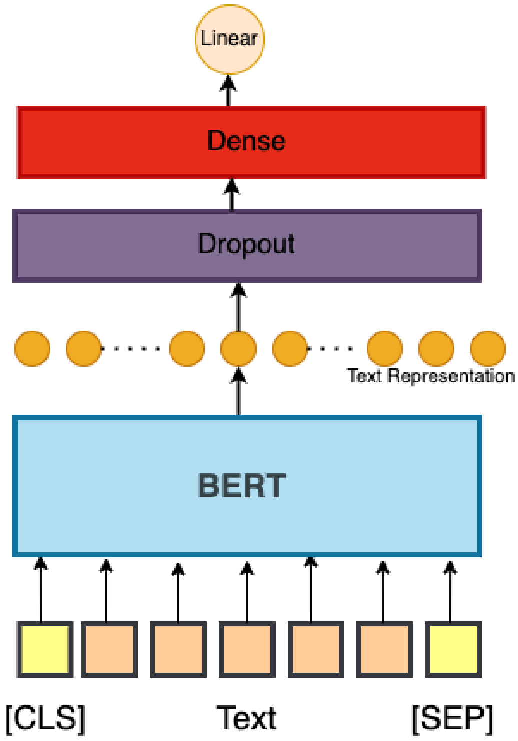


Figure 5: Model Architecture for Subtask B

Parameter	Value
Model	bert-base-uncased
Max sequence Length	256
Batch Size	8
Learning rate	2e-5
Weight decay	Linear
Momentum	0.9
Optimizer	AdamW
Epochs	5
Loss	Mean Squared Error

Table 4: Experimental Setup for Subtask B

score of 44.200% which is substantially above a naive majority class baseline of 39% and comparable to the baseline presented by the task organizers.

3.2 Subtask B

For Subtask B, the submissions were evaluated using Spearman's rank correlation coefficient(SRCC) which compares the predicted plausibility ranking overall test instances with the gold ranking. We ranked 5th with an SRCC of 0.25200.

4 Error Analysis

After examining the predictions from the submitted model, we saw that the model struggled significantly in distinguishing between neutral and either of plausible/implausible clarifications as there's a very slight difference between them. This problem increases further when we created a feature vector from the same sentence with changed filler word as it leads to a very slight change in the vector, hence leading the model confused to distinguish between individual classes having almost similar feature distribution. The performance can also be attributed to the distribution of the labels in development set and our submitted model might have overfitted to the development set, leading to a further decrease in performance in the test set.

5 Conclusion

We developed a system to classify the clarification made on instructional text into varying levels of plausibility using a transformer based language model with a limited attention span only taking a limited context around the filler. The recent advancement in transformer based models, such as BigBird (Zaheer et al., 2020) and Longformer (Beltagy et al., 2020) which can take up longer context into consideration are more preferred for a task like this.

References

Talita Anthonio, Irshad Bhat, and Michael Roth. 2020. wikiHowToImprove: A resource and analyses on edits in instructional texts. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5721–5729, Marseille, France. European Language Resources Association.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Irshad Bhat, Talita Anthonio, and Michael Roth. 2020. Towards modeling revision requirements in wikiHow instructions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8407–8414, Online. 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*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Michael Roth and Talita Anthonio. 2021. UnImplicit shared task report: Detecting clarification requirements in instructional text. In *Proceedings of the 1st Workshop on Understanding Implicit and Underspecified Language*, pages 28–32, Online. Association for Computational Linguistics.

Michael Roth, Talita Anthonio, and Anna Sauer. 2022. SemEval-2022 Task 7: Identifying plausible clarifications of implicit and underspecified phrases in instructional texts. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297.