DuluthNLP at SemEval-2022 Task 7: Classifying Plausible Alternatives with Pre–trained ELECTRA

Samuel Akrah & Ted Pedersen
Department of Computer Science
University of Minnesota
Duluth, MN 55812 USA
{akrah001, tpederse}@d.umn.edu

Abstract

This paper describes the DuluthNLP system that participated in Task 7 of SemEval-2022 on Identifying Plausible Clarifications of Implicit and Underspecified Phrases in Instructional Texts. Given an instructional text with an omitted token, the task requires models to classify or rank the plausibility of potential fillers. To solve the task, we fine–tuned the models BERT, RoBERTa, and ELECTRA on training data where potential fillers are rated for plausibility. This is a challenging problem, as shown by BERT-based models achieving accuracy less than 45%. However, our ELECTRA model with tuned class weights on CrossEntropyLoss achieves an accuracy of 53.3% on the official evaluation test data, which ranks 6 out of the 8 total submissions for Subtask A.

1 Introduction

Instructional texts (e.g., How To Guides) describe how to accomplish a given goal and are integral to our daily lives. One popular source is WikiHow[1] which is an online platform that allows users to collaborate to create and maintain such guides. This kind of documentation must be clear, and if it is not then this is a key reason that prompts revisions of underspecified instructions in WikiHow (Anthonio et al., 2020a).

One important problem for NLP is to determine if a given instructional text is in need of clarification or revision. SemEval-2021 Task 7 (Roth et al., 2022) extends this problem by requiring models to score five possible fillers based on how well they can plausibly fit a given context. Task 7 includes two subtasks. Subtask A classifies the possible fillers as IMPLAUSIBLE, NEURAL, and PLAUSBILE. Subtask B requires systems to rank the fillers on a scale of 1 to 5, where a higher score means more plausible. We only participated in Subtask A and used a variety of BERT-based methods.

[1] https://www.wikihow.com

2 Task Data

The training, development and test data were supplied by the organizers of SemEval–2022 Task 7 (Roth et al., 2022). The dataset is based on the WikiHowToImprove Corpus (Anthonio et al., 2020b) which consists of edits of 2.5 million sentences from WikiHow. The authors show that edits are primarily made to clarify instructional texts and that the distinction between older and revised versions of sentences can be modelled computationally.

Each instance in the data is divided into an Article title, a Section header, and a Sentence nested between a Previous and Future context. Each instance also includes five potential fillers, each of which is annotated as IMPLAUSIBLE, NEUTRAL, or PLAUSIBLE, which serves as the basis of Subtask A. There is a also a plausibility score of 1 to 5 which is the basis of Subtask B (which we did not participate in).

Table 1 shows 3 training example templates made up of the concatenation of the Previous context, the Sentence with the blank to be filled, and the Future context. Each template is filled with each of the five possible fillers to generate the training examples (5 per template). We start with 3,995 training templates where the filler is not specified. We create an instance for each of the five possible fillers for each template, giving us a total of 19,975 training examples. We remove extraneous content such as bullet points and numbers in order to make the examples more readable. In a later approach, we highlight each filler in the generated instance with a special "[filler]" token, a step that yields further performance gains on the development data (Table 3) but achieves no corresponding gains on the official test results.

3 Methodology

We experimented on three large pre–trained language models for Subtask A, including BERT,

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1062 - 1066

Previous context	Sentence	Followup context
State what you have contributed to the company. By doing it this way, it is going to show that you have done your job and been an asset, thus the raise is well-deserved. *	**P:** If you believe your [...] of time working at this company warrants a raise or promotion, say that as well. **Fillers: A. continuity B. window C. abundance D. length E. appreciation**	It is best to tell them all the reasons you believe you deserve this increase.
An all weather strategy often keeps you always afloat compared to one planned for normal market behavior. Planning for a failure is always better than failing to plan	**P:** Uncertainties of [...] can be classified into four levels **Fillers: A. public opinion B. future markets C. the future D. this sort E. their futures**	Level one gives a fairly clear view of the future, and an inkling of what to expect.
Since wikis are often volunteer-driven projects, wikigifts can go a long way in showing someone how much you appreciate their efforts. Find or create awards specific to that wiki.	**P:** On wikiHow, for example, you can Make Award Templates on [...] and post them on people's talk pages. **Fillers: A. books B. facebook C. graph D. wikiHow E. earth**	Publicize that you gave the wikigift.

Table 1: Training Example Templates for SemEval–2022 Task 7. Each possible replacement (filler) for the omitted token [...] must be ranked as PLAUSIBLE (blue), NEUTRAL(orange), or IMPLAUSIBLE (RED).

RoBERTa and ELECTRA. BERT (Bidirectional Encoder Representational from Transformers) (Devlin et al., 2018), is a Transformer-based language model trained using Masked Language Modeling (MLM) to predict the masked tokens based on the surrounding context. In MLM, a given percentage of the tokens of an input sequence is masked, and BERT is tasked to predict the orginal tokens. With the MLM approach, BERT was able to produce good results when transferred to downstream NLP tasks, becoming the new benchmark for other pre–trained models. The authors of the ELECTRA paper (Clark et al., 2020), however, note that the MLM approach only learns from the masked tokens (about 15%) of any given example, thus requiring substantial compute resources to train a language model using MLM.

The next model we used was RoBERTa, which is essentially a replication of BERT (Liu et al., 2019) which adjusts key hyperparameters and uses larger amounts of data during pre–training. RoBERTa improves upon BERT when trained longer, using larger mini–batches over more data.

Similarly, we experimented with ELECTRA (Clark et al., 2020), a pre–trained model that uses Replaced Token Detection as a pre–training objective. This distinguishes real inputs from plausible but synthetically generated ones coming from a small masked language model. The authors argue that ELECTRA improves compute efficiency during pre–training, and can match or exceed the performance of BERT and its variants when fine–tuned on downstream tasks.

The ELECTRA model includes two Transformer models, a generator and a discriminator. The generator emulates a small Masked Language Model by predicting the original token of a masked-out token. Like the Masked Language Model in BERT, some samples of the input sequence to the generator are replaced with [MASK]. The predicted results of the generator are fed as inputs to the discriminator.

For each token in the sequence, the discriminator predicts whether it is the original or the generated one. This means that the discriminator is able to learn from all input tokens for any given example, with the model loss calculated over all the tokens. This is what sets ELECTRA apart from BERT, and a major reason for ELECTRA's greater compute efficiency.

ELECTRA addresses a drawback of Masked Language Models, which is that the masked tokens are only used during pre–training and are omitted during fine–tuning. This pre–train fine-tune mismatch contributes to a loss in performance. Replaced Token Detection, in which ELECTRA distinguishes between real tokens and their plausible fakes, is easily transferable to fine–tuning. This is particularly true for Subtask A of SemEval-2022 Task 7, which requires a model to classify how fillers will plausibly fit an omitted token.

4 System Description

This section introduces DuluthNLP's approach which relied on BERT, RoBERTa, and ELECTRA.

Hyperparameter	Value
Learning Rate	4e-5
Adam ϵ	1e-8
Optimizer	AdamW
Learning rate decay	Linear
Weight Decay	0
Batch Size	16
Train Epochs	10

Table 2: Hyperparameter Values for Fine-Tuning.

Model	Accuracy
ELECTRA with class weights	0.556
RoBERTa with class weights	0.552
BERT with class weights	0.522
ELECTRA	0.443
BERT	0.441
Logistic Regression	0.348
Random Guessing	0.267
RoBERTa	0.177

Table 3: Experimental Results on Development Data.

We discuss our fine–tuning process, and then how class weights were tuned.

4.1 System Description

We first fine–tune BERT on Subtask A. Using our pre–processed dataset as inputs, we build and train a classifier on top of the BERT model to learn how plausible each filler fits the blank in each sentence.

Using the BERT-base uncased tokenizer on our inputs, we then train our BERT model using the Adam Optimizer with a linear scheduler with warmup; a CrossEtropy Loss with adjusted class weights; and a learning rate of 4e-5 for 10 epochs (see Table 2). We train our model twice, once without class weight tuning, and a second time with tuned class weights.

We used these same hyperparameters for fine-tuning RoBERTa (Liu et al., 2019) and ELECTRA (Clark et al., 2020). As our experimental results will show, the most accurate results were obtained with ELECTRA.

We used the HuggingFace PyTorch implementations of the BERT, RoBERTa, and ELECTRA (Wolf et al., 2019). We fine-tuned our models using 2 Nvidia Quadro RTX 8000 GPUs.

4.2 Class Weights

Class imbalances for classification tasks are often caused by imbalances in the dataset. However, for Subtask A the training data is reasonably balanced and includes 7339 (36%) PLAUSIBLE labels, 7162 (36%) NEUTRAL labels, and 5474 (27%) IMPLAUSIBLE labels.

While the task training data does not have significant imbalances, our model predictions on the development set initially skewed towards NEUTRAL. Over 52% of all the predictions were NEUTRAL, as shown in Table 4.

Our model corrects these imbalances by applying class weights that penalize NEUTRAL labels. This helped to reduce predictions for NEUTRAL labels to 28%, as shown in Table 5. This is at least closer to the actual distribution of 18% NEUTRAL in the development data and helps to improve accuracy.

The selection of optimal weights was based on random search, which has been shown to be more efficient for parameter optimization than grid search (Bergstra and Bengio, 2012). To achieve this, we initially defined a 3-tuple list of random weights, and for each tuple, we set the class weights for the CrossEntropy Loss function and trained our model. From the list, we selected the class weights with the best performance for further tuning.

5 Experimental Results

In this section we present the results of our models on both the development data and the test data as used in the official evaluation scored by the task organizers.

We used the Logistic Regression model from scikit-learn[2] as a baseline method. It incorporates binarized Ngram counts (Wang and Manning, 2012).This obtained an accuracy of 34.8% on the development data for Subtask A.

As shown in Table 3, accuracy on the development data without class weights were lower. The RoBERTa model, in particular, achieved accuracy of 17.7%, the lowest among the three language models, and even lower than random guessing.

However, with tuned class weights, all three of the pre–trained models achieved accuracy above 50%. ELECTRA and RoBERTa obtained nearly identical scores of 55.2% versus 55.6%. We decided to use ELECTRA as our official evaluation

[2] https://scikit-learn.org/

		Predicted			Total	%
		0	**1**	**2**		
Actual	**0**	249	580	140	969	38%
	1	53	255	135	443	18%
	2	45	456	587	1088	44%
	Total	347	1291	862	2500	
		14%	52%	34%		

Table 4: Confusion Matrix with ELECTRA Before Weight Tuning on Development Data. The labels (0,1,2) refer to (IMPLAUSIBLE, NEUTRAL, PLAUSIBLE) respectively. Accuracy is 42.4%.

		Predicted			Total	%
		0	**1**	**2**		
Actual	**0**	332	340	297	969	38%
	1	70	128	245	443	18%
	2	60	221	807	1088	44%
	Total	462	689	1345	2500	
		18%	28%	54%		

Table 5: Confusion Matrix with ELECTRA After Weight Tuning on Development Data. Accuracy is 50.7%.

method because of its very consistent performance with the class weights adjustment and its somewhat lower energy consumption as compared to other large language models.

Our official results on the Subtask A evaluation data for Task 7 Subtask A were 53.3% with our ELECTRA model with class weights. The top ranked system in the task obtained accuracy of 68%. DuluthNLP ranked 6 among 8 systems.

6 Error Analysis

The classes predicted by our model on the development data prior to the official evaluation were skewed to NEUTRAL, as discussed earlier. We observed this with various different pre–trained models including BERT, RoBERTa, and ELECTRA with various different hyperparameter settings. Despite our best efforts the DuluthNLP system never reached accuracy above 45%.

We addressed this by adding class weights to CrossEntropyLoss function used in our models. When we assigned class weights of [1.5, 0.03, 0.7] for the IMPLAUSIBLE, NEUTRAL, and PLAUSIBLE labels, the wrongly predicted scores for the NEUTRAL label reduced to 28%, as shown in Table 5.

What is curious, though, is the difficulty in correctly classifying the IMPLAUSIBLE label. This represents 38% of the actual labels but is 14% of the predicted labels. Even after class weights are set as described above, only 18% of the labels are predicted to be IMPLAUSIBLE, which is still a difference of 20% from the actual IMPLAUSIBLE labels.

We achieved further performance gains on the devset across all the models by highlighting the filler in each data instance with a special "[]" symbol, and this forms the basis for our results in Table 3. This approach did not distribute well over the test data, however.

7 Ethical Considerations

The training of large language models has a dark side: demands for large amounts of compute power and the corresponding energy consumption. Training BERT with the Masked Language Model requires a lot of computational resources.This raises concerns over the accessibility, cost, and environmental impact of such methods (Bender et al., 2021). Whilst we experimented with three BERT-variants, we sought to limit model fine–tuning to the base models (Bert-base, RoBERTa-base, and ELECTRA-base), which require less compute resources than their larger versions. The ELECTRA model, which we used for our official evaluation test results, is computationally efficient, which partly informed our choosing it over the other models for use as our official method during the evaluation stage.

The accuracy of our models are, at best, a little above 50%. This means that roughly half of any of these predictions may be wrong. This is clearly not accurate enough to deploy in a real setting without potentially causing harm. It is easy to imagine the negative impacts of automatically clarified instructions that prove to be inaccurate.

Similarly, the test data and the training set are from the same sample distribution, and we cannot guarantee that our model will achieve similar results for any out-of-distribution test data. In other words, our model, reliant as it is on the contextual representations provided by the pre–trained models, cannot perform well on a completely new task

or distribution.

We relied on large language models which were trained on very large corpora. Such text may include stereotypes and biases which are then carried over into the resulting model (Bender et al., 2021). This locks the model to older, less–inclusive understandings that may not reflect more modern views of gender, race, or other questions of identify. To minimize the potential harms of such misrepresentations it would be best if candidate predictions were verified by a human editor.

However, if models like these were deployed and used to auto–suggest revisions to WikiHow, they may constrain the choices of human editors (Miller and Record, 2017). This could create a mindset that uncritically accepts the framed options of the model as legitimate (Alfano et al., 2018). The reviewer may then abandon their own edits because the auto–suggestion seems to provide an answer. One way to minimize such risk is the deployment of Reflection Machines (Cornelissen et al., 2022), a decision support system that will compel users to give reasons for accepting or rejecting suggestions from the model.

Acknowledgements

The authors would like to thank the organizers for the opportunity to participate in SemEval-2021 Task 7. We are also grateful to the three anonymous reviewers for their thoughtful comments and feedback. Finally, we would like to thank Dr. Alexis Elder for her contributions regarding the ethical considerations of this work.

References

Mark Alfano, J. Adam Carter, and Marc Cheong. 2018. Technological seduction and self-radicalization. *Journal of the American Philosophical Association*, 4(3):298–322.

Talita Anthonio, Irshad Bhat, and Michael Roth. 2020a. 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.

Talita Anthonio, Irshad Bhat, and Michael Roth. 2020b. 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.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

N. A. J. Cornelissen, R. J. M. van Eerdt, H. K. Schraffenberger, and W. F. G. Haselager. 2022. Reflection machines: Increasing meaningful human control over decision support systems. *Ethics and Inf. Technol.*, 24(2).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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. *CoRR*, abs/1907.11692.

Boaz Miller and Isaac Record. 2017. Responsible epistemic technologies: A social-epistemological analysis of autocompleted Web search. *New Media & Society*, 19(12):1945–1963.

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.

Sida Wang and Christopher Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94, Jeju Island, Korea. Association for Computational Linguistics.

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