# Stanford MLab at SemEval 2022 Task 7: Tree- and Transformer-Based Methods for Clarification Plausibility

**Thomas Yim**[*], **Junha Lee**[*], **Rishi Verma**, **Scott Hickmann**, **Annie Zhu**,
**Camron Sallade**, **Ian Ng**[†], **Ryan Chi**[†], and **Patrick Liu**[†]
Stanford University
{yimt, junhalee, rishirv, hickmann, anniezhu}@stanford.edu,
{camron, iyhn8192, ryanchi, pliu1}@stanford.edu

## Abstract

In this paper, we detail the methods we used to determine the idiomaticity and plausibility of candidate words or phrases into an instructional text as part of the SemEval Task 7: Identifying Plausible Clarifications of Implicit and Underspecified Phrases in Instructional Texts. Given a set of steps in an instructional text, there are certain phrases that most plausibly fill that spot. We explored various possible architectures, including tree-based methods over GloVe embeddings, ensembled BERT and ELECTRA models, and GPT 2-based infilling methods.

## 1 Introduction

The internet is filled with instructional texts from websites like wikiHow that detail how to perform a variety of tasks (from tying a bow tie to building a deck of stairs). With this increase of quantity and use of instructional texts, it has become increasingly important for them to be clear and unambiguously worded. To this end, we evaluate whether lightweight and neural models are capable of detecting which phrases most plausibly fit into a given series of instructions.

The current revision process requires that a reader potentially identify something wrong about the content, and they report it to the website for alterations. A system such as the one described could automatically identify candidates that are unlikely to exist and report them for human verification.

In this paper, we describe our lightweight and transformer-based models that rank the plausibility of candidate phrases given some previous context.

## 2 Background

### 2.1 Task Setup

The data for this task was provided by SemEval Task 7 (Roth et al., 2022). This dataset is an augmented version of WikiHowToImprove (Anthonio et al., 2020), which consists of 2.7 million sentences and their revision histories extracted from the instructional website WikiHow. The dataset extracts over 4000 sentences and revision that represent clarifications of the original text. Each sentence is masked and presented with 5 possible fillers that may represent a clarification. The article title, subsection, previous context, and future context are also provided. Each filler is annotated with a plausibility class label (either IMPLAUSIBLE, NEUTRAL, or PLAUSIBLE), whose prediction is the basis of Subtask A, and a plausibility score on a scale from 1 to 5, whose prediction is the basis of Subtask B.

The plausibility of a given filler is highly dependent on the context. For example, the clarification *birthday* is annotated as implausible for the sentence

*2. Send a ______ card. Even if you're able to have the conversation in person, it's worth sending a card.*

given the previous context

*1. Ask in person whenever possible. Receiving an invitation to be a groomsman is exciting.*

as the previous context implies a wedding.

Given this, we simply concatenate each of the provided data inputs into a single string as input for our models.

## 3 System overview

### 3.1 Word Embeddings

We first implemented a GloVe-based method (Pennington et al., 2014). To embed all the words, we used the Python Natural Language Toolkit (nltk) library to handle tokenizing all words. These words were pre-processed using a Punkt sentence tokenizer that can handle stripping punctuation from boundaries that would not affect the semantics of the phrase or sentence. We then fit various classical ML models, as listed in Section 3.2, to make

[*] Co-first authors.
[†] Co-senior authors.

*Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1067 - 1070
July 14-15, 2022 ©2022 Association for Computational Linguistics

predictions on the word vector inputs.

### 3.2 Lightweight models

- **Ridge regression** is a classification algorithm that minimizes the residual sum of squares.
- **Random forest** is a supervised learning technique that ensembles independent decision trees to yield a result.
- **Gradient Boosting** is a technique that ensembles a number of weak learners (typically decision trees) and optimizes based on a differentiable loss function.
- **Discriminant analysis** is a generative learning algorithm that assumes the data is distributed according to a Gaussian distribution.
- **Multilayer Perceptron** is a multi-layer artificial neural network.

### 3.3 BERT-like models

BERT is a language representation model first introduced in 2018 that has achieved state-of-the-art results in NLP experiments (Devlin et al., 2018). The model follows a multi-layer transformer-based encoder architecure. It leverages bidirectional self-attention, which enables it to learn context from both preceding and following sentences. Furthermore, it was trained on language modeling tasks – predicting masked tokens from context – which makes it an ideal model for this task. We fine-tune the BERT model to learn the plausibility of various possible fillers.

To do so, we input both the masked sentence with context information (chosen out of resolved pattern, article title, section header, previous context, and follow-up context) and the possible filler, with each element separated by a special separator token. We then train a shallow neural network on top of BERT to recognize the plausibility or implausibility of the filler based on BERT's encoded representation.

In our data set, we used the pre-trained bert-base-uncased tokenizer to prepare the words for the model. In our BERT Class, we used the Sigmoid activation function and added dropout to prevent over-fitting of the model. We froze the first 8 layers of the BERT model and then trained using Cross Entropy Loss, an Adam optimizer function, and a learning rate scheduler. As the model was trained, the program kept track of the best models with the highest performance accuracy. It would check with each epoch and save the model if it improved.

Alongside BERT, we also experiment with other similar BERT-based language models, like ALBERT (Lan et al., 2019) and RoBERTa (Liu et al., 2019). ELECTRA uses the same underlying model as BERT, but through a different pre-training mechanism. This approach corrupts the input by replacing tokens with alternatives, rather than masking it, and trains the model to determine which tokens were corrupted versus part of the original sentence (Clark et al., 2020). This makes ELECTRA another ideal candidate for determining the plausibility of certain fillers. Each of these models was fine-tuned and evaluated in the same manner as the original BERT.

The ELECTRA model of the highest accuracy was achieved using the ELECTRA-small discriminator, two linear layers, a hyperbolic tangent activation function, and a dropout rate of 0.5 before each linear layer. The context information used was the previous and follow-up context.

### 3.4 Infilling by Language Modeling

This approach is taken from (Donahue et al., 2020). Training data for this model was created by randomly masking words or phrases in a body of text and fine-tuning the GPT-2 model on those masked sentences. We combined the Article Title, Section Header, Previous Context, Sentence, and Follow-up Context into a single string and added an infill mask token where we are predicting the word or phrase. Then, given the surrounding the context, the model returned a set of logits and softmaxed probabilities that ranked the probability of all possible tokens. To handle multitoken words such as "jalapenos" ([474, 282, 499, 28380]), the logits were summed and the probabilities were mutliplied for each individual token. There were additionally phrases like "your hands" ([14108, 2832]) that were multi-token as well.

For the example "How to Store Jalapenos," the sentence needing clarification was "Make sure to wear latex gloves when handling jalapenos or wash [INFILL-WORD] thoroughly after handling." As follows are the options and their returned logits: "your hands": -6.1058 "the jalapenos": -70.2685 "the sun": -22.6745 "the floor": -19.1986 "your underwear": -18.6225. The model accurately determined "your hands" to be more probable than the other options like "the sun", "the floor", and "your underwear" which make little sense in this context. However, "the jalapenos", which received a medium plausible score of 3.0 in the training data was found to be extremely improbable with this model due to the high number of tokens. This was a flaw that was pervasive throughout the usage of this model and is why we decided to continue with the BERT and ELECTRA ensemble models instead.

After the logits were calculated for all the possible infills in the training data, we trained a Ridge linear model to convert from logits to our 1-5 scoring scale. Unfortunately, because of the inaccurate probabilities generated by multi-token words and phrases, there appeared to be no correlation between logits and their labeled scores, leading the Ridge model to predict around 3.33 as a baseline score for most words.

## 4 Experimental setup

We merge the provided train and dev sets, perform random 75:25 splits of the merged data to use for training and validation. We noted little difference in performance between different random splits.

Predictions were evaluated on accuracy for Subtask A and Spearman correlation for Subtask B. Although we also calculated other metrics such as macro-averaged F1 for Subtask A and mean squared error for Subtask B, we standardized on accuracy and Spearman correlation for consistency in comparing results. In particular, since the classes were close to balanced for Subtask A, using accuracy was not a big issue in overfitting to certain classes.

## 5 Results

As shown in Table 1, here was some variation in dev set performance between the lightweight models that we experimented with; in particular, Linear and Quadratic Discriminant Analysis performed better than the other models. However, even these performances are very low, achieving a maximal dev set accuracy of only 0.389 with Linear Discriminant Analysis.

After switching to BERT-like models, we generally achieve a significant improvement in classification accuracy. In particular, ELECTRA achieves the highest accuracy across all of our models, with a dev set accuracy of 0.465. An exception is RoBERTa, with which we achieve an accuracy close to that of Linear Discriminant Analysis.

The ILM-based models ultimately failed to improve upon the accuracy of the lightweight models. While the ILM approach might initially seem to be the most promising given the task, it seems that the model's ability to generate the top few most likely options did not correlate with its ability to compare relatively more unlikely potential infills. The model was also pre-trained on a stories database, which may not reflect the context appropriate for instructional texts like WikiHow articles.

For the regression subtask, we only experiment with BERT with a regression head. We find that BERT achieves a Spearman correlation of 0.149 on the dev dev set.

Our final evaluation results on both subtasks are shown in Table 2. Surprisingly, our evaluation scores are slightly higher than our development scores.

## 6 Conclusion

BERT-like models are the highest performing models for this type of instructional clarifications task. Because BERT has been trained to predict masked tokens, it is naturally better at finding words or phrases that most plausibly fit with the surrounding context. Simple GloVe word embedding models were unable to learn to the requisite complexity of this meta-linguistic task and were unable to break the level of 0.389 accuracy. Meanwhile, ILM-based approaches seemed promising, but in practice failed to accommodate phrases or long words. This method is more effective at generating text and not necessarily determining how plausible an infill sounds in that context.

In terms of future work, we believe that our system may be improved by the usage of large pretrained models such as SpanBERT (Joshi et al., 2019), which are trained using tasks involving multi-token span prediction, which may be more fit to the given task. Despite our lack of results, we believe that the infilling approach is still promising, and hope that it can be adapted to this task going forward.

## Acknowledgements

This research effort would not have been possible without the support of Stanford ACMLab. The authors thank Michael Roth, Talita Anthonio, and Anna Sauer for organizing SemEval 2022 Task 7: Identifying Plausible Clarifications of Implicit and Underspecified Phrases in Instructional Texts.

| Model | Accuracy |
|---|---|
| K Nearest Neighbors | 0.347 |
| ILM | 0.354 |
| Ridge | 0.357 |
| Random Forest | 0.363 |
| MLP Classifier | 0.365 |
| Hist Gradient Boosting | 0.366 |
| Quadratic Discriminant Analysis | 0.373 |
| RoBERTa | 0.387 |
| Linear Discriminant Analysis | 0.389 |
| BERT | 0.447 |
| ELECTRA | **0.465** |

Table 1: Dev set performance (Subtask A)

| Subtask | Method | Result |
|---|---|---|
| A (Accuracy) | ELECTRA | 0.496 |
| B (Spearman R) | BERT | 0.194 |

Table 2: Test set results.

Additionally, the authors would like to recognize Google Colaboratory for their free compute services.

## References

Talita Anthonio, Irshad Bhat, and Michael Roth. 2020. wikiHowToImprove: A resource and analyses on edits in instructional texts. In *LREC*.

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

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*.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. *CoRR*, abs/2005.05339.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

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*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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.