SI 630 Homework 3:
Data Annotation and Large Language Models

Part 0 Due: Thursday February 24, 11:59pm
Part 1 Due: Wednesday March 9, 5:30pm
Part 2 Due: Wednesday March 16, 5:30pm

Last updated: Feb 22 (version 1.0)

1 Introduction

People love to ask questions in social media—there are entire sites like Quora and StackOverflow dedicated to asking and answering questions. But what makes for a helpful response? Helpfulness can come in many forms: a story, a code snippet, a favorite book, sound advice, or more. In this homework, we'll look specifically at this notion of a helpful answer. You will design annotation guidelines to rate replies to answers using a 5-point Likert scale from (1) Not Helpful to (5) Very helpful.

Homework 3 is a two-part group homework focused on teaching you (1) designing annotation guidelines and measuring annotation quality and (2) building classifiers based on large language models. You'll go through the full Machine Learning pipeline from initial unlabeled data to training classifiers. This data curation process is often overlooked as a simple step, but as a practitioner in the field, if you have to annotate your own data (or hire people to label for you), you'll need these skills to ensure your eventual classifier performs well.

This homework has the following learning goals:

1. Learn how to iteratively craft annotation guidelines
2. (Optional) Learn how to use annotation tools
3. Learn how to compute rates of inter-annotation agreement
4. Learn how to adjudicate disagreements during data labeling
5. Learn how to use the Huggingface `Trainer` infrastructure for training classifiers
6. Improve your skills at reading documentation and examples for advanced NLP methods
7. Learn how to use the Great Lakes cluster and access its GPUs
8. Understand how annotation agreement and guideline-divergence influence classification performance

This homework is multi-part because we will use the annotations produced by all teams to train and analyze your classifiers. Roughly, the first two weeks will be spent developing your guidelines and annotating. Once all teams have finished, we will release the larger set of annotations and teams will develop their classifiers. Teams can get started doing the classifier development earlier using just their own data and then train the classifier on the final data.

2 Class-wide Technical Report

As you will see in this assignment, annotation is challenging and annotated data—especially, *quality* annotated data—is incredibly valuable. Based on the number of teams we have, SI 630 will have produced a new and interesting resource for NLP models: A rich set of thousands of responses labeled for their helpfulness *and* multiple detailed annotation guidelines for different interpretations of helpfulness, along with which guidelines were used for which labels.

As a class, I would like us to publish our collective resource on helpful replies as a technical report on ArXiv with you all as authors (opt-in) where we share different guidelines and our annotations to contribute to the research community. You would not need to contribute anything beyond your anonymized guidelines and annotations to be an author. Writing and analysis contributions would move you up in the author order. If there's interest we could even try to publish the technical report as well, which has been done for past course projects, e.g., Kenyon-Dean et al. [2018][1] You all are doing amazing and it's important to realize that you have the ability to advance the state of NLP and data science in general with the skills. The technical report would be a way to document your contribution.

3 Forming a Team

You will design your initial annotation guidelines in teams. Therefore, the very first step that is needed is to form a team of 2 to 3 people. **There are no single-person teams or teams of four or more students.** Register your team at this link: `https://forms.gle/wzjHvSLJc2TBVRdN7`. Once registered, you will receive your team's data items to annotate.

Note that this form also asks if your group would allow its anonymized guidelines and annotations to be included in the class technical report. While totally optional, what you are creating as a part of the class has real value and I would encourage you to share your knowledge with the community, which you'll get authorship credit for. If you have any questions, please feel free to reach out to David on Piazza.

■ **Problem 1.** Register your team using this link `https://forms.gle/wzjHvSLJc2TBVRdN7` to receive your specific annotation assignments.

4 Part 1: Annotating Data

4.1 Designing Annotation Guidelines

Helpful replies can take on many different forms. How should you design your guidelines to label each reply. What makes a comment a 1 versus a 2 or a 4 versus a 5? We've included a *large* sample of questions and replies from Reddit to get you started thinking about what qualities to look for.

As a first step, each team member should write up their own guidelines for what makes a helpful response. Your guidelines should clearly define how any annotator should judge helpfulness along

[1] `https://aclanthology.org/N18-1171/`

each point in the 5-point Likert scale.[2] It is often helpful to include examples and descriptions of what is (or is not) an example of each scale point. Your individual guidelines should be *at least* one page, single spaced using one inch margins. You will likely end up writing more than one page and writing your thoughts/notes now will be helpful when putting together your group's guidelines.

Be sure to draft your guidelines independently without talking to your teammates. These independent thoughts on how to annotate are incredibly useful for framing and will make for a good discussion with your team since you will have each thought about how to design guidelines and where to make distinctions in helpfulness. You will have your own definition of helpfulness so writing it down (as guidelines) before meeting with the group will be important for expressing yourself.

■ **Problem 2.** (15 points) Each team member drafts their own annotation guidelines. Each person should upload their own guideline to Canvas. Your guidelines should not have any identifying information in it (e.g., your name or uniqname).

4.2 Designing Annotation Guidelines

Once each team member has drafted their own annotation guidelines, you will meet as a team and collaboratively construct a combined annotation guideline. Try annotating some of the data together with it to see where your scales differ or what example you should move. Does it cover all the cases you think it should? Use your insight from this process to revise the team's annotation. We strongly recommend repeatedly randomly sampling a small number of examples, rating them separately using the team's guidelines, and then discussing disagreements to revise the guideline further.

Your annotation guidelines should provide clear criteria that an annotator could use to determine where any comment goes on the rating scale. You should all agree on the label distinctions. We highly recommend an iterative process where group members meet, independently annotate a few items using the group's guidebook, and then reconvene to discuss any disagreements and how the guidebook could be improved or refined to prevent those in the future.

We recommend providing guidelines that are *at least* four pages.[3]

■ **Problem 3.** (15 points) Draft annotation guide that will be used as a group. Upload the group's guidelines to Canvas. Because we will redistribute these documents, your guidelines should not have any identifying information in it (e.g., your names or uniqnames). Specify the group members via Canvas group.

[2] All teams will use this 5-point scale, so you must adopt this in your own task.

[3] If you are initially not sure how a guidebook can be that long, do a series of group annotations to see disagreements and then write clear criteria/qualities and examples to help annotators avoid those disagreements. Documentation is key!

4.3 Annotating Your Data

Once your team has decided on a set of annotation guidelines, each team member will use the guidelines individually to rate your assigned items. Do your best to follow the guidelines and feel free to make notes on the annotation process (or guidelines) during annotation. You'll use these notes later when you recommend updates.

You are welcome to use any annotation software you want. Perhaps the simplest option is to load the data into a Google Sheet creating a new column called "Rating" which will make it easier to aggregate these ratings later. You can then work through each item and easily download the annotated data as a .csv for use later in training. If you want to get more sophisticated, you can also turn on Data Validation (Data -¿ Data Validation in the menu) and provide a list of options to annotate. Another option is to use a fancier annotation tool like Potato (`https://github.com/davidjurgens/potato`) or any of the others mentioned in Week 7 in class (e.g., LightTag, Labelstudio, Prodigy, etc.). In general, you should find a tool and workflow that maximizes annotation throughput while helping you prevent mistakes.

■ **Problem 4.** (15 points) Individually label the data using the group's guidelines. You are required to follow the Academic Honesty code and not discuss your annotations with each other.

■ **Problem 5.** Once finished annotating, collect your group's annotations into a single .csv file in the format like the following

```
id,annotator,rating
s987u2a,user1,3
s987u2a,user2,2
s987u2a,user3,3
g902a0f,user1,1
g902a0f,user2,1
g902a0f,user3,2
```

and upload your file to Canvas. Your file should have no additional columns and should load easily using `pd.read_csv("yourfile.csv")`. Failure to do this will result in lost points.

We will eventually distribute these files and anonymize the usernames so you do not need to anonymize them yourselves.

4.4 Measuring Inter-Annotator Agreement

Once you have finished your annotation, it's now time to measure quality! You will measure annotation agreement in two ways: (1) Pearson's correlation r and (2) Krippendorff's α. For the latter, you should use a Python package and we recommend either `https://github.com/pln-fing-udelar/fast-krippendorff` or `https://github.com/grrrr/krippendorff-alpha`.

■ **Problem 6.** (5 points) Perform the following agreement analyses:

1. Compute r and the compute α using the *ordinal* and *nominal* level of measurements for the group member's annotations and report them (three scores total).

2. In 2-3 sentences, comment on the difference (if any) between your group r and the α scores. Which is higher and what do you think this means?

3. In 2-3 sentences, comment on the difference (if any) between your group's ordinal and nominal α scores. Which is higher and what do you think this means? Which one should you use in practice to measure agreement in this setting?

■ **Problem 7.** (5 points) Once *all* the annotations are released, compute the agreement of all the other annotators on your group's items. In a 2-3 sentences, comment on the difference (if any) between your group's and other students' agreement and what you think is the cause. We recommend looking at other groups' guidelines at this point.

4.5 Examining Disagreements

Once all of the data is released, we'll look for places where your group disagreed with other students' decisions. Conveniently, each item will have been annotated by two groups, each with their own guidelines. What led to the disagreements? Was it differences in guidelines? Differences in annotators' interpretations? Fatigue? Lexical ambiguity? There are *many* reasons and this part of the assignment will have you grappling with the challenge of defining truth and understanding why some annotators might disagree.

For each item, compute the mean score for the group's ratings and the mean score for another group of students. We'll treat the mean as the group's estimate, recognizing that there are other ways to chose a final label. Examine the 10 instances that have the biggest absolute difference in mean rating between your group and the other group. Often this type of review process is helpful for identifying gaps in your guidelines.

■ **Problem 8.** (5 points) In your report, include the text from each of these 10 replies and what the ratings were for each group. Describe why you think the two were different, ideally pointing to differences in the two group's guidelines. After a discussion with your group members, adjudicate the rating and decide what is the final "true" rating, examining all the evidence. If you changed your score, in one sentence, state what led you to do this.

■ **Problem 9.** (5 points) Based on your analysis of annotator agreement and the disagreements and the different annotation guidelines, provide *at least* specific improvements that could be made to your own guidelines to improve annotator agreement. Improvements should be a series of bullet points, each describing in at least 1-2 sentences the changes that would be made to increase annotator agreement or correct for gaps and edge cases.

5 Part 2: Recognizing Helpful Answers

Part 2 will have you developing classifiers using the very powerful Hugging Face library.[4] Modern NLP has been driven by advancements in Large Language Models (LLMs) which, like we discussed in Week 3, learn to predict the word sequences. Substantial amounts of research have shown that once the models learn to "recognize language," the parameters in these models (the weights in the neural network) can quickly be adapted to accomplish many NLP tasks. We'll revisit the task from Homework 1: classification.

Most of the work on LLMs has focused on BERT or RoBERTa, which are large masked language models. These models have been shown to generalize well to a variety of new tasks. However, the models have billions of parameters which can make them slow to train. As a result, another branch of work has focused on making these language models smaller—can we distill a smaller model from what the larger model has learned, or, start with a smaller model and use different techniques to have it learn the same thing. The "BERT family" of models all have the same structure of inputs and outputs (and core neural architecture: the Transformer network) but vary in how they're trained and how many neurons they use. Thankfully in the `huggingface` API, all of these models can be accessed in the same way; the end-user (you) just needs to specify the model family and the parameters to plug in. For this assignment we will use a much smaller but nearly-as-performant version of BERT, `https://huggingface.co/microsoft/MiniLM-L12-H384-uncased`, to train our models.

This part of the assignment will have you using Great Lakes cluster, which has provided free GPUs for us to use.[5] You will need to use Great Lakes for this assignment to be able to train the deep learning models. We have made it an explicit learning goal as well, so that you are comfortable with the system for Homework 4 and using the cluster for your course projects.

To train your classifier, you have a few options. The simplest is to treat each scale point as a class and do multi-class classification. However, as you might notice, many of the ratings are not exactly all whole numbers so you would need to round, which loses some information. Further, if the answer is scored a 5 and the model predicts a 4, that prediction is just as wrong as a prediction of 1 in the classification setting. Therefore, a slightly more aligned model would be a *regression model*. Regression models frequently use a different loss, commonly Mean Squared Errors, unless how we used Cross-Entropy Loss (or Binary Cross Entropy Loss) in previous homework. There are many other options too, depending on how exotic you want to get. You are free to use whatever model and loss you want and we'll see how well they perform.

■ **Problem 10.** If you don't have an account on Great Lakes, request one now at `https://arc.umich.edu/login-request`

For training your LLM-based classifier, we'll use the `Trainer` class[6] from `huggingface` which wraps much of the core training loop into a single function call and support many (many) extensions and optimizations like using adaptive learning rates or stopping your model if it converges to avoid overtraining. The class's documentation has many options, which can be a it overwhelming at first. We recommend searching out examples and blog posts on how to use the class; finding

[4] `https://huggingface.co/`

[5] `https://arc.umich.edu/greatlakes/user-guide/`

[6] `https://huggingface.co/docs/transformers/main_classes/trainer`

and interpreting these kinds of informal documentation is a learning goal for this assignment. As you progress in your technical career, honing your skills as quickly finding and making use of these (likely, as you did for StackOverflow) will be critical for using state of the art NLP models

■ **Problem 11.** (15 points) Develop your code using `huggingface`'s `Trainer` class to train a classifier or regressor to predict the helpfulness rating of an answer. Submit your code to Canvas and report your score on the development dataset in your team's report.

For this homework, we'll be trying the most stringent form of evaluation. You are welcomed and encouraged to do extensive testing of your model on the development data. However, you will get only one shot at predicting the test data. The motivation for this setup, is that it is the most rigorous test of how well your model generalizes to unseen data.

■ **Problem 12.** (10 points) Use your trained `huggingface` model to predict the helpfulness score on the test dataset and submit it to Kaggle. Your grade will depend, in part, on how well your model performs. **You only get one submission to Kaggle, so make it count!**

5.1 Annotation Evaluation via Classification

Once you have the code for training the model ready, we can also put it to use to examine how annotators and guidelines might have influenced the model performance. Specifically, you'll perform *annotator ablation tests* where we hold out the annotations of some groups of annotators and see how (1) the model performance changes and (2) in the validation set, whether model predictions are more or less similar to those annotators' labels.

You'll do the following procedure for each group $g_i \in G$:

- Remove all the annotations by any annotator in g_i from the training data
- Create ground truth data by averaging the scores of all remaining annotators in the training data
- Train your helpfulness prediction model on these aggregated scores
- For the development data, split into three sets: (a) replies without any annotators in g_i, (b) replies to items with annotators in g_i and using their annotations (c) the same replies as *b* but using the annotations by the individuals *not* in g_i. The first set will measure how well the model is doing in general. The difference between (b) and (c) is whether the model is more able to predict the scores of group g_i (even though it wasn't trained on this data) or whether the model's predictions are closer to everyone else's predictions
- For the three development sets, take the average score of the annotators as the ground truth.
- Estimate your model's performance using Pearson's correlation on each of the validatation subsets.

■ **Problem 13.** (10 points) Make a bar plot of the correlation scores of each group for the three evaluation subsets in the format shown in Figure 1. In a paragraph, describe what you see and

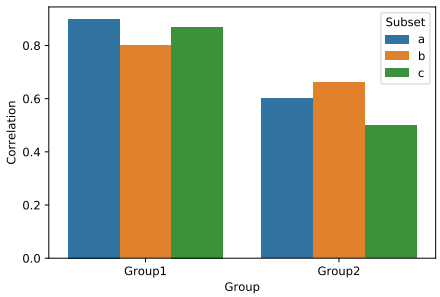

Figure 1: An example of the format of the figure comparing classifier performance when specific group's annotations are held-out and evaluated, described in Section 5.1. This figure is shown for two groups; your figure should show all groups. If space is too tight, you can split all the groups into separate figures with multiple groups per subfigure.

discuss the relative impact of each team's annotations. For example, did some group's annotations cause the model to perform worse? Were some group's labels closer to the model's predictions? In your discussion, review the relevant groups' guidelines and describe whether specific instructions in their guidelines might be the cause of these differences in scores/performance.

6 What to submit

All group-based submissions should be done by forming a group in the Canvas assignment like you do for the lecture notes.

- Each person submits their own guidelines to Canvas
- One person submits the group's annotation guidelines to Canvas
- One person submits the group's annotations
- One person submits the group's implementation of the helpfulness prediction model
- One person submits a single PDF document with the group's answers and plots to all questions in this document.
- One person should submit their group's predictions for the test set on Kaggle (NB: you only get one upload!)

7 Academic Honesty

Unless otherwise specified in an assignment all submitted work must be your own, original work. Any excerpts, statements, or phrases from the work of others must be clearly identified as a quotation, and a proper citation provided. Any violation of the University's policies on Academic and Professional Integrity may result in serious penalties, which might range from failing an assignment, to failing a course, to being expelled from the program. Violations of academic and

professional integrity will be reported to Student Affairs. Consequences impacting assignment or course grades are determined by the faculty instructor; additional sanctions may be imposed.

Annotating in your group in a non-independent way is considered grounds for violation of Academic Integrity and will receive a zero for that part of the assignment.

References

Kian Kenyon-Dean, Eisha Ahmed, Scott Fujimoto, Jeremy Georges-Filteau, Christopher Glasz, Barleen Kaur, Auguste Lalande, Shruti Bhanderi, Robert Belfer, Nirmal Kanagasabai, Roman Sarrazingendron, Rohit Verma, and Derek Ruths. Sentiment analysis: It's complicated! In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1886–1895, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1171. URL https://aclanthology.org/N18-1171.