Title: Safety for Conversational AI

Goal: Evaluate and develop datasets, methods, and systems for the detection and mitigation of safety issues for conversational AI.

Possible approaches:

- Detection of unsafe inputs
- Analysis of system outputs

Possible subtopics:
Suggested in [1, 4]

- Finance (Nandita, detection)
- Legal (Syed, detection)
- Sensitive topics, e.g. politics and or religion [4]
- Conspiracy theories and misinformation, e.g. COVID 19
- ~~Self-harm, mental health~~ (too sensitive, ethical approval difficult/impossible)

Methods

Detection of unsafe inputs:

1. **Search for a source for your positive class examples** (i.e. the unsafe questions). In [3], we used the subreddit r/AskDocs because it (i) has a lot of data (ii) we looked at some of the posts and they seemed to contain suitable questions.
2. **Download the positive data** from pushshift [5]. Nandita has code for this. You will want only the post titles as these are nice and short and likely to contain questions.
3. **Download the negative data** (i.e. safe questions not on your topic). In [3], we just took data from a mixture of random subreddits.
4. **Filter out non questions**. You can use the Dialogue Act Classifier to identify questions https://www.nltk.org/book/ch06.html 2.2 Identifying Dialogue Act Types
5. **Verify your data** take a sample of each class and check that your positive examples are on topic (and are questions?) and that your negative examples are safe. Report the result e.g. as a percentage. Is it good enough? If not, what can you do to improve the data collection process?
6. **Preprocess data**
    a. Balance classes - you probably want roughly the same number of pos and neg examples
    b. Add labels to your positive and negative datasets, 1 for positive, 0 for negative. So X_pos = [unsafe_question_1, unsafe_question_2, …, unsafe_question_n]; y_pos = [1, 1, …, 1], X_neg = [safe_question_1, safe_question_2, …, safe_question_n], y_neg = [0, 0, …, 0]
    c. Mix and shuffle your data.
7. **Feature extraction.** You now need to convert the words in your data to numbers. A common way is tf-idf https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

8. **Divide data into training and test sets.** https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html What is a good proportion? 90%/10%? 80%/20%? Or you could try k-fold validation https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html
9. **Establish baseline(s)**. What happens if you guess the labels in the test set at random? Or always the most common label (if your data is not balanced)?
10. **Classification experiments**. Choose which classifiers to use (logistic regression, knn, svm, mlp, …) https://scikit-learn.org/stable/supervised_learning.html
11. **Evaluation**. Compare the classifier predictions with the "true" labels. If your data is balanced, you can report accuracy. If not, or to get more insights, you might need some other metrics like F1 score https://scikit-learn.org/stable/modules/model_evaluation.html

Analysis of system outputs:

An alternative is to test the way conv. systems respond to the questions (you could do this for a topic that has already been chosen by another student for input detection).

1. **Get prompting data**. Follow steps 1 and two above, or ask a student working on that topic for their data. (You only need positive examples for this).
2. **Choose some systems to test**. E.g. DialoGPT on HuggingFace https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
3. **Collect system responses**.
4. **Devise an annotation scheme.** How are you going to label the responses? Binary (safe/unsafe)? Or ordinal (on a scale like [3])
5. **Annotate the data.** Read the responses and label them according to your scheme
6. **Verification.** Can you get someone else (another student?) to label your data or a subset of it? Calculate inter-annotator agreement https://towardsdatascience.com/inter-annotator-agreement-2f46c6d37bf3
7. **Extension: Classification**. How well do ML classifiers do at recognising unsafe output (as above)

Ensuring you finish

- What is the minimal version of the project? Is there anything you could skip or leave until the end?
- For each stage, what's the smallest viable version? For example, for classification, you could run just one classifier.

Extensions

What can you do to create a better project and get a better grade?

For each stage, think about how you could extend it and provide a more thorough analysis. Some ideas:

Detection:

1-2 For a more challenging task, you could add more unsafe topics. I.e. multiclass classification of financial vs legal vs neither. Or you could try for finer grained classes: e.g. different types of legal question
5. Verify a greater proportion of the data. Or get another person to verify some data and report inter-annotator agreement
6. What happens if you experiment with lower casing, stemming, lemmatising, removing stopwords, different n-grams
7. Experiment with tf-idf parameters.
8. Compare different ways of splitting data
9. Report more baselines
10. Compare more classifiers. Very advanced: Fine tune a transformers model https://huggingface.co/docs/transformers/main/en/index

System outputs:

1. add more unsafe topics.
2. Add more systems
3. Collect more responses
4. More complex annotation scheme e.g. ordinal labels
6. Have more people annotate the data
7. Do the classification experiments

Related literature:

[1] Emily Dinan, Gavin Abercrombie, A. Bergman, Shannon Spruit, Dirk Hovy, Y-Lan Boureau, and Verena Rieser. 2022. SafetyKit: First Aid for Measuring Safety in Open-domain Conversational Systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4113–4133, Dublin, Ireland. Association for Computational Linguistics.

[2] Cercas Curry, A. & Rieser, V. (2018) #MeToo: How Conversational Systems Respond to Sexual Harassment. *Proceedings of the Second ACL Workshop on Ethics in Natural Language Processing*

[3] Abercrombie, G. & Rieser, V. (2022) Risk-graded Safety for Handling Medical Queries in Conversational AI. In *Proceedings of the Annual Meeting of the Asian Association for Computational Linguistics*.

[4] Xu, Jing, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. "Recipes for safety in open-domain chatbots." *arXiv preprint arXiv:2010.07079* (2020).

[5] Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. Proceedings of the International AAAI Conference on Web and Social Media, 14(1):830–839.