

P2 - Response Type Classification in Discussions

Phase I DUE: Thu, Sep 24th at noon

Phase II DUE: Sun, Oct 4th at noon

Programming language to use: Python3

Learning Objective

In this assignment, you will learn feature engineering through a classification task.

Background

Interactions through question-answering play an important role in discussions. Through questioning, askers may want to elicit information (e.g., wh-questions), clarify situations (e.g., closed-ended questions), or even make a point (e.g., rhetorical questions). However, how a question is responded does not necessarily align with the intent of the asker. For example,

Asker: How could you possibly know that?

Responder: I am not claiming this is a statement of fact, it is an opinion.

In the example above, the asker implies a point “no one could know that” with a rhetorical question, however, though generally no response is expected for a rhetorical question, the responder attacked the point the asker makes.

In this assignment, you will need to model the relations between a question and response and classify the interaction into different types.

Dataset

Phase I and phase II tasks are based on question-response pairs extracted from online discussion forums. Context (precedent and subsequent sentences of a question in the same post) of questions will be given for your reference.

Phase I: Annotation

We have predefined several types of responses as in the table below. (we use Q/R to denote question/response respectively).

<i>response type</i>	<i>explanation</i>	<i>examples</i>
----------------------	--------------------	-----------------

irrelevant	The response is (null) or the response is not related to the question.	e.g. Q: Your religion is morally disgusting and doesn't make any logical sense. Do you know what science is? R: As an example we can agree that stoning kids because they don't obey is wrong.
agreed	The response shows an assertion with the claim implied by the question	e.g. Q: Well, who wants to clean their junk before they throw it away. R: That's right.
answered	The response takes the question literally and tries to answer the question (although sometimes the intent for this type of answering is attacking)	e.g. Q: If I burn my hand, does telling me fire is hot help me? R: Yup. e.g. Q: What if Mary is carrying mace, but it doesn't help her? R: This solution implies you ignore the risk, but carry self-defense in case bad things happen. e.g. Q: Should all the blacks in the state move out? R: I can't make decisions for other people.
attacked	Attacked, or say rebutted. The response attacks or satirize the question (on its assumption, reasoning, etc.) without answering the question	e.g. Q: Do you see the difference? R: You see the difference. e.g. Q: What if the mugger chooses Mary because a previous mark looked too intimidating? R: My argument is on opportunistic crimes, not premeditated ones.

You will be distributed a list of pairs in the following format (please ignore the columns of ids):

<i>Given. from asker</i>			<i>Given. from responder</i>	<i>your label</i>	<i>your naming/ justification of 'other' type</i>
precedent	question	subsequent	response	type	justification (if type == 'other')

Notes about the format and extracted context:

- If a previous comment is directly quoted in the current comment as a context of a question, the quoted comment is marked with ">" as a block quotation format.

- If a phrase or partial content is quoted from a previous comment, the quoted content is marked with “<quote>...</quote>” as an inline quotation format.

After Phase I, the annotated samples will be aggregated, and the gold label of response type will be chosen with the majority criterion for each sample. The aggregated samples will be released to you for the Phase II task. **Note: Don't change the name of the files you have received via email.**

Phase II: Feature Engineering and Classification

The annotated response types are aggregated and released to you for the Phase II task.

Task:

- Extract the baseline features
- Propose two additional features that you think would contribute to type classification
- Vectorize the features and classify response types
- Evaluate classification performance
- Write a report. Include a link to the lexicon in the report if any is proposed as additional features

Baseline features (Include these features as your baseline model):

- word/sentence embeddings
- Part-of-speech tags

Some ideas for additional features:

- Indicating words or expressions in questions or responses such as “even”, “can't you”. Include a reference to the lexicon you employed or a justification for why you pick those expressions as linguistic clues.
- sentiment analysis of questions and responses.

One challenge of this task is that you need to find a way to “combine” the vectorized representation of questions and responses.

For example, to use word/sentence embeddings, a straight-forward way is to learn a sentence embedding of a question and a response, concatenate the two vectors, and use a simple classifier such as [SVM](#). The same applied to part-of-speech tags via [NLTK](#). For high-level features such as sentiment scores, be creative on how to incorporate the scores for a question and its response.

Model:

You should experiment with multiple classification models (such as SVM or neural-network-based classifiers) and choose the best to report results for. You are required to report the results for the following feature sets:

- set 1: only baseline features
- set 2: baseline features + **two** additional features (that you proposed)

Evaluation:

- accuracy
- precision, recall, and F1 score

Based on the performance of the model on the two feature sets, analyze why your proposed additional features contribute or not to type classification.

Submission:

- Report with descriptions and screenshots of your code and evaluation metrics (**only pdf version will be graded**), comparison and conclusion of your methods.
- Source code in .ipynb format with output of required evaluation.
- If you will submit a .py source code, a ReadMe file to explain the compilation and execution of your program, make sure the code will be executed successfully and when it is necessary to comment / uncomment code when switching pre-trained models and others.