Adversarial crowdsourcing-based human feature selection

Discussed before the meeting

Motivations

1. Feature selection is important for analyzing performance in interactive machine learning tasks
2. Prevent workers from having bias (preconceptions) of your task
    - Workers might try to guess what you mean by "features" on straightforward tasks
3. We want to calibrate judgments
    - Opponent's response gives signal on worker's performance

Method

1. Design a game-like interface for workers and associate two opposing group
2. Pilot study
3. Deploy to crowd
4. Compare with naive approach of directly asking

Meeting minutes

Motivation/tasks

1. General goal:
    - A lot of work on elicitating labels, can we design a game that elicitate both labels and features that should be considered if we are classifying the documents into the labels? (why the data show up: like what mentioned in Autocoder)
    - Test their intuition/human intelligence without additional context (how the algorithm is behaving)
2. Possible ways of phrasing the task
    - "Can people do this" task: Understand if users are reasonable source for figuring out the useful features that make a document dinstinguishable
    - Comparison tasks: What do you think is important to a machine algorithm v.s. what is important to humans

Input/Domain setting

1. Binary text classification: easier than multi-classfiers
2. Corpus
    - IMDB movie review: balanced, long documents
    - Twitter data: more sparse and shorter documents (might choose this?)
    - Spam/non-spam: less balanced distributoin

Deliveries

Possible data to collect from the experiment

1. Labels: human judgement (could be two judgements if in the adversarial case)
    - validity of the data, data without labels
    - If people can label things correctly
2. Features relevant to the label
    - The minimal information needed to stop the guessing person to arrive at the same judegement

Possible evaluation methods

1. Compare the features collected with those from automated models
    - TF-IDF: generally important
    - Info gain: Corpus distribution
    - Human gathered set: Domain knowledge
2. Compare to a system that explicitly elicitate features (or: does the phrasing and the mechanism of the task affect how workers behave?)
    - Ask questions like "which features do you think are important?" (this could feel like an ambiguous question and can affect how people think.)

Gamification setting

(View it as an interative design rather than formal research so we don't need IRB consensus.)

Additional benefits of gamification

1. Engaging
2. Automatic evaluation on the feature collected

Alternative 1: Adversarial: reduce people's chance of classifying correctly

Approach

1. Randomly pair peoples to be groupmates, each group have a feature selection person and a guessing person
2. Feature selection person:
    - Label the document
    - Select features/keywords to delete to prevent the oponents to classify the document correctly
3. Oponent/guessing person: partially blocked documents, classifiy the documents (can answer "I can't guess").

Problem

1. Can people be paired up efficiently (same working time, etc.)?
2. How to punish selecting unnecessary words?
3. Pick one word v.s. pick sets of words?

Alternative 2: One person filling in the blocked words in the sentence

Approach (metaphor: million dollar Pyramid)

> The game features two contestants, each paired with a celebrity. Contestants attempt to guess a series of words or phrases based on descriptions given to them by their teammates.

1. Start with a incomplete sentence with important sentimental words blocked
2. See (1) what words a person might want to view first, (2) what is the final set we collect

Problem

1. We are gathering the "position" of the words they are interested in, not the actual word, as they are already blocked.

Alternative 3: reverse the meaning of the words

Approach

1. Twitter sentiment example: reverse the highly emotional words to see if the sentiment of the whole sentence reverse.

Problem

1. Does this still work in other tasks, e.g., spam/non-spam?

Potential difficulties

1. Motivation for interactive ML: only to interact with models when your data is not sufficient/your problem is unique (certain things not available)
    - "I believe in the data rather than the person": people's intuition are wrong
2. Syncronization: a person obscure and get back guesses (automate the procress?)
3. Feedback: read-world rewards, points, etc.
4. Design the game to be fun+visually compelling+works for the task v.s. end up like a data-entry problem
    - Machansim
    - The choice of the data
5. Making games can be time consuming

Important elements in the proposal/presentation

1. Well-defined underlining objective/structure: what exactly it is that you are trying to get out (again refer to the autocoder thing)?
2. Well-defined milestons showing efforts toward making the game fun + ways to evaluate it
    - The mapping between the objective and the game
    - Paper prototyping + evaluation the prototyping (see how other papers evaluate; could publish to gaming websites and see how many stars they get)
    - Implementation
3. e.g., 4 different designs of sketched prototypes that have different underlying mechanisms, so we can deliver to collect feedbacks

References

1. mustache, ESP game (a human-based computation game developed to address the problem of creating difficult metadata.)

    - Keyword elicitation v.s. just bag-of-words for text
2. Edith Law, Luis
    - X-with-a-Purpose system

    - Search War: A Game for Improving Web Search: query the webpage, what keywords matter in that webpage in terms of the query result
    -

3. Highlight features instead of documents (???)
4. Image attention
    - One person game
    - Start with a black image, click a part to reveal a small sub-image to identify the image
    - The fewer the click, the higher the reward
5. Kisskissban

    > In a KKB game, one player, the blocker, competes with the other two collaborative players, the couples; while the couples try to find consensual descriptions about an image, the blocker's mission is to prevent the couples from reaching consensus.

6. Pace games
    - keep up with things that are coming by
    - Sort them based on the keywords, etc.
    - Could feel like constructing a decision tree
7. Autocoder (Simone Stumpf) explanatory debugging: supporting end-user debugging of machine-learned programs
    - The context is different each time.
    - Qualitative research
    - Transcripts from interviews
    - Encode the complains occur in the transcript: person complaining about the teacher, etc.
    - Build a bag of classifiers by giving them examples from the text
    - Similar to annotating images for entity recognition
8. designing games with a purpose