Project Template

Emily Tseng

Abstract

- This document describes the expected style, structure, and rough proportions for your final project write-up.
- While you are free to break from this structure, consider it a strong prior for our expectations of the final report.
- Length is a hard constraint. You are only allowed max **8 pages** in this format. While you can include supplementary material, it will not be factored into the grading process. It is your responsibility to convey the main contributions of the work in the length given.

1. Introduction

Example Structure:

- What is the problem of interest and what (high-level) are the current best methods for solving it?
- How do you plan to improve/understand/modify this or related methods?
- Preview your research process, list the contributions you made, and summarize your experimental findings.

In therapeutic and caregiving contexts, a patient's felt sense of *alliance* with a care provider can make or break their treatment. This is particularly important in remote caregiving contexts like online therapy, where the vast majority of the connection between the patient and the caregiver takes place entirely over text-chat.

In this work, we explored the use of neural representations of conversational dynamics in forecasting alliance between patients and therapists.

2. Background

Example Structure:

- What information does a non-expert need to know about the problem domain?
- What data exists for this problem?
- What are the challenges/opportunities inherent to the data? (High dimensional, sparse, missing data, noise, structure, discrete/continuous, etc?)

3. Related Work

Example Structure:

- What 3-5 papers have been published in this space?
- How do these differ from your approach?
- What data or methodologies do each of these works use?
- How do you plan to compare to these methods?

NLP and therapy.

Conversational forecasting. Methodologically, our work is an adaptation of the Conversational Recurrent Architecture for ForecasTing (CRAFT), a framework integrating generative pre-training with a supervised fine-tuning model to achieve improved predictive ability on conversation-level attributes, e.g., whether an online conversation will derail into personal attacks (Chang & Danescu-Niculescu-Mizil, 2019).

4. Dataset

We consider a dataset of therapy transcripts and associated patient outcomes from Talkspace, an online therapy platform. Due to the highly sensitive nature of therapy, we put significant effort into respecting patients' privacy and autonomy. All represented patients gave informed consent for the use of their data in research, and transcripts were anonymized by Talkspace before they were handed to our research team. Our study protocol was approved by our institutional IRB.

In total, our dataset consists of 5.7M messages exchanged between patients and therapists, representing 11,233 patients' full courses of treatment. 1,906 therapists are represented, with an average of 9 patients per therapist (stdev

[FIXME: TODO]). [FIXME: A smattering of more descriptive statistics: length of pt message, length of therapist message, number of pt / th messages per conversation, length of treatment, burstiness of treatment...]

Outcome annotations are provided in the form of patients' responses to surveys issued approximately every 3 weeks. In total, 13,742 WAI scores were provided by 6,702 patients. Patients provided an average of 2.1 WAI scores (range 1-24, stdev 2.2), and the overall distribution of scores skewed strongly towards the positive end of the spectrum (more strongly allied) [FIXME: check and then insert figure].

5. Model

Example Structure:

- What is the formal definition of your problem?
- What is the precise mathematical model you are using to represent it? In almost all cases this will use the probabilistic language from class, e.g.

$$z \sim \mathcal{N}(0, \sigma^2) \tag{1}$$

  But it may also be a neural network, or a non-probabilistic loss,

$$h_t \leftarrow \mathrm{RNN}(x_t, h_{t-1})$$

  This is also a good place to reference a diagram such as Figure 1.

- What are the parameters or latent variables of this model that you plan on estimating or inferring? Be explicit. How many are there? Which are you assuming are given? How do these relate to the original problem description?

We define a conversation C as a variable-length sequence of n utterances, $C = \{u_1, ..., u_n\}$. Utterances are variable-length sequences of tokens w, and thus $u_n = \{w_1, ..., w_{M_n}\}$, where M_n is the length in tokens of utterance n.

Problem definition. Given a therapy exchange $C = \{u_1, ..., u_n\}$, we aim to predict y_n, the WAI score provided by the patient at utterance u_n.

Generative component. Following Chang et al. 2019, we adopted for our generative component the hierarchical recurrent encoder-decoder (HRED) architecture proposed in Sordoni et al. 2015 and Serban et al. 2016. Built to model high-level conversational context, including temporal structure and dependencies between consecutive sequential inputs, HREDs are uniquely suited for conversational forecasting tasks.

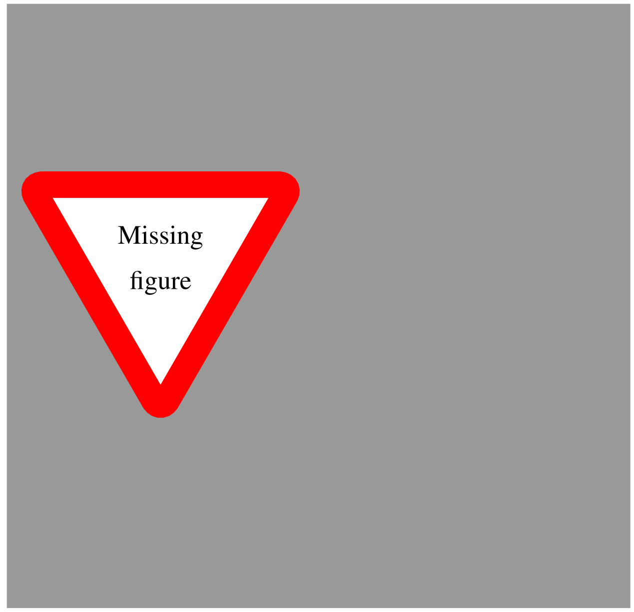

Figure 1. This is a good place to include a diagram showing how your model works. Examples include a graphical model or a neural network block diagram.

HREDs are comprised of three component recurrent neural networks (RNNs): an utterance encoder, a conversation encoder, and a decoder. First, the *utterance encoder* generates for each utterance a semantic vector representation via its hidden state $h^{enc} \in \mathbb{R}^{d}_{enc}$, where d_{enc} is the desired dimension. For each token w_m in utterance n of length M, the encoder updates its h^{enc} like so:

$$h^{enc}_m \leftarrow f^{RNN}(w_m, h^{enc}_{m-1}) \tag{2}$$

The utterance encoder's hidden state at the last step, h^{enc}_M, in theory represents an embedding for the entire utterance. Following Serban et al. 2016, h^{enc}_0 is initialized as the zero vector $\mathbf{0}$, and following Chang et al. 2019, we use the GRU (Cho et al., 2014) as our nonlinear gating function f^{RNN}.

Next, the *conversation encoder* uses the hidden states from each consecutive comment in a sequence of length N to produce an embedding h^{con}_n for the conversation up to the utterance at that point (u_N):

$$h^{con}_n \leftarrow f^{RNN}(h^{enc}_{M_n}, h^{con}_{n-1}) \tag{3}$$

The conversation encoder also initializes its hidden state h^{con}_0 with the zero vector $\mathbf{0}$, and also uses the GRU as its nonlinearity. We denote the dimension of h^{con} as d_{con}.

The *decoder* uses the embedded conversational context h^{con}_n to generate a response to utterance n. Following Sordoni et al. 2015, it does this by first initializing its own hidden state $h^{dec} \in \mathbb{R}^{d_{dec}}$ using a nonlinear activation of h^{con}_n:

$$h^{dec}_0 = \tanh(Dh^{con}_n + b_0) \tag{4}$$

Where $D \in \mathbb{R}^{d_{dec} \times d_{con}}$ projects the context embedding into decoder space, and $b_0 \in \mathbb{R}^{d_{dec}}$. The decoder then updates its own hidden state for each response token using the following recurrence:

$$h_t^{dec} \leftarrow f^{RNN}(w_{t-1}, h_{t-1}^{dec}) \tag{5}$$

The decoder then produces the next token in its response by producing a probability distribution over words from h_t^{dec}:

$$w_t = f^{out}(h_t^{dec}) \tag{6}$$

In our implementation, we use a simple feedforward network for f^{out}, including a softmax.

Predictive component.

Parameters.

6. Inference (or Training)

- How do you plan on training your parameters / inferring the states of your latent variables (MLE / MAP / Backprop / VI / EM / BP / ...)
- What are the assumptions implicit in this technique? Is it an approximation or exact? If it is an approximation what bound does it optimize?
- What is the explicit method / algorithm that you derive for learning these parameters?

Algorithm 1 Your Pseudocode

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

7. Methods

- What are the exact details of the dataset that you used? (Number of data points / standard or non-standard / synthetic or real / exact form of the data)
- What are the exact details of the features you computed?
- How did you train or run inference? (Optimization method / hyperparameter settings / amount of time ran / what did you implement versus borrow / how were baselines computed).
- What are the exact details of the metric used?

8. Results

- What were the results comparing previous work / baseline systems / your systems on the main task?
- What were the secondary results comparing the variants of your system?
- This section should be fact based and relatively dry. What happened, what was significant?

9. Discussion

- What conclusions can you draw from the results section?
- Is there further analysis you can do into the results of the system? Here is a good place to include visualizations, graphs, qualitative analysis of your results.
- What questions remain open? What did you think might work, but did not?

10. Conclusion

- What happened?
- What next?

References

Chang, Jonathan P and Danescu-Niculescu-Mizil, Cristian. Trouble on the horizon: Forecasting the derailment of online conversations as they develop. In *Proceedings of EMNLP*, 2019.

Cho, Kyunghyun, Van Merriënboer, Bart, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Serban, Iulian V, Sordoni, Alessandro, Bengio, Yoshua, Courville, Aaron, and Pineau, Joelle. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Table 1. This is usually a table. Tables with numbers are generally easier to read than graphs, so prefer when possible.

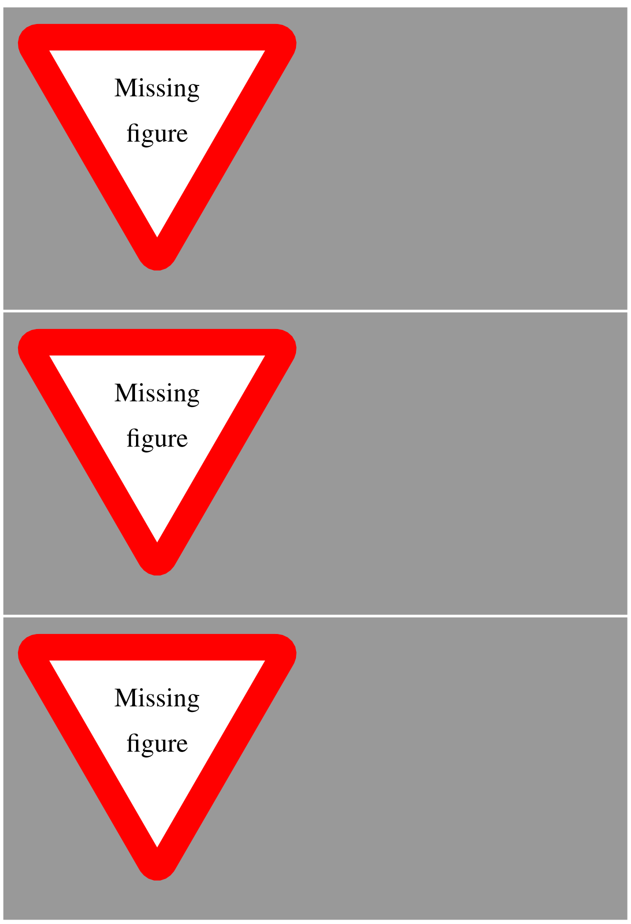

Figure 2. Visualizations of the internals of the system.

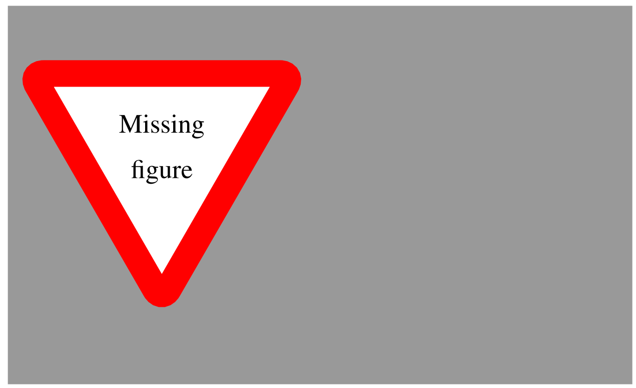

Table 2. Secondary table or figure in results section.

Sordoni, Alessandro, Bengio, Yoshua, Vahabi, Hossein, Lioma, Christina, Grue Simonsen, Jakob, and Nie, Jian-Yun. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 553–562, 2015.