interpretable methods for doc alignment in dialogue

Anonymous ACL submission

Abstract

TBD

1 Introduction

In many customer-facing dialogue applications, customer service interactions must follow a set of guidelines for safety, which have a natural sequential order. If a customer is locked out of their account and requests a password reset, the agent must first verify that the customer is indeed the owner of the account. This if-then structure is common in guidelines (Chen et al., 2021).

All agents, whether human or robot, must follow safety guidelines. As a result, safety guidelines are often written in natural language. Natural language guidelines also allow for zero-shot generalization to scenarios that may be new to an agent, but described similarly in the guidelines to familiar scenarios.

Our goal is to train dialogue agents that not only follow a set of guidelines, but justify their actions by pointing to the guidelines. This allows others to verify their actions, and whether the guidelines have been followed.

We propose a generative model of dialogue, that justifies decisions by aligning to a guidelines, utilizes the sequential structure of guidelines, and does not require supervision.

Experiments show that our model is accurate, intepretable, and works at a range of supervision levels.

We present results on three datasets, ranging over a variety of guideline styles. In ABCD, the guidelines are given to us Chen et al. (2021). In SGD, we write the guidelines ourselves, using the generative model to aid development. In doc2dial, we show that our method works for alignment to general document-guided dialogue as well.

2 Related work

The adaptation of large languge models to task-oriented dialogue has allowed for impressive results in zero-shot generalization, where models are tested in scenarios that they have not previously seen (). The key idea behind this success is the use of a natural language interface: specify scenario-specific details using natural language, and take advantage of the generalization abilities of large language models.

3 Problem setup

Our goal is to, given an observed task-oriented and guideline-grounded dialogue x between a customer and agent, justify the actions of the agent by aligning them to natural language guidelines z.

4 Method

We propose a generative model of dialogue that justifies its actions by aligning to the guidelines.

The model first chooses a document in the guideline $z \sim p(z)$, then generates the dialogue $x \sim p(x \mid z)$. This yields the joint distribution $p(x, z) = p(x \mid z)p(z)$.

We perform training by optimizing the log marginal likelihood

$$\log \sum_z p(x, z). \tag{1}$$

We perform inference online via Bayes' rule:

$$\operatorname*{argmax}_z p(z \mid x) = \operatorname*{argmax}_z p(x \mid z)p(z). \tag{2}$$

Why not break down alignments at the turn-level? We found that using a document to generate only the next agent turn resulted in poor unsupervised accuracy (degeneration to a uniform distribution). Additionally, we found that many single turns were well-explained by a large number of different documents. It is these two points that led us to consider generating full dialogues given a single document, so that document must explain multiple turns at once. This is because documents in guidelines share many common actions, and it is the sequencing of these actions that distinguishes them. Therefore modeling the whole dialogue allows the model to take into account full sequences of actions. Note that this is only for documents. We will perform lower-level alignments at the turn-level, while keeping document selection at the full dialogue level.

4.1 Wake-sleep training

We perform additional experiments with an inference network to speed up training over full marginalization. We propose the following objective:

$$\log \sum_z p(x, z) - KL[p(z \mid x) || q(z \mid x)]. \quad (3)$$

In order to speed up training, we make the following approximations:

$$\log \sum_z p(x, z) \approx \log \sum_{z \in Z'} p(x, z)$$

$$KL[p(z \mid x) || q(z \mid x)] \approx KL[\tilde{p}(z \mid x) || \tilde{q}(z \mid x)]$$

$$\tilde{p}(z \mid x) = \frac{p(x, z)}{\sum_{z \in Z} p(x, z)}$$

$$\tilde{q}(z \mid x) = \frac{q(z \mid x)}{\sum_{z \in Z} q(z \mid x)},$$

so that $\tilde{p}, \tilde{q}$ are only normalized over $Z' = \text{argtopk} q(z \mid x)$. Note that this topk operation is not differentiable.

Why not VAE training? We found VAE training with the usual ELBO required a baseline and achieved worse accuracy even with a leave-one-out baseline. We hypothesize that this is because the wake-sleep objectiven naturally uses a regret baseline, while the VAE baseline can be interpreted as the advantage. Additionally, the VAE objective optimizes reverse KL, which may be worse than the forward KL in this case, since we an uncalibrated q may ruin training by becoming overconfident about an incorrect z.

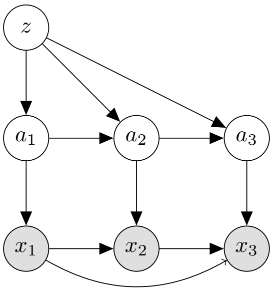

Figure 1: Graphical model for full document and span alignments.

5 Parameterization

We parameterize $p(x \mid z)$ with a sequence to sequence model such as BART. The prior $p(z)$ is a uniform distribution. The inference network $q(z \mid x) \propto \langle \text{enc}(x), \text{enc}(z) \rangle$ encodes x, z with a transformer such as RoBERTa.

6 Span alignments

Assert no multi-hop.

7 Experiment 1: Document classification from observed dialogue + action output

7.1 Research question

Can we do document classification in document-driven dialogue with as few document labels as possible?

7.2 Experiment

Unsupervised document classification with a generative model of dialogue given document. document accuracy $z|x$ right before first agent action. The first agent action will be something like pulling up the customer's account, at which point the agent should be following a document. The agent will always know what the correct document is before taking an action, but they will definitely know what the correct document is right before they take an action.

7.3 Models

- Skyline: supervised

$$p(z \mid x) \propto \langle emb(z_{label}), BERT(x) \rangle$$

model	acc
Skyline supervised $p(z\|x)$	90.65
Baseline lexical	34.06
Approx marg w/ Z^*	80.18
Full marg w/ Z	74.2

Table 1: Results for document classification with a generative model at the first agent action in a conversation.

- Baseline: lexical BM25
- Approximate marginalization with Z^*: $\log \sum_{z \in Z*} p(x \mid z)p(z)$
  - Z^* = {true z^*, 3 hard lexical negatives based on z^*, 3 random negatives}
  - uniform $p(z)$
  - BART $p(x \mid z)$
  - Inference via Bayes' rule: $\text{argmax}_z\, p(x_{1:t}|z)$ where t is the index of the first agent action.
- Full marginalization over Z: $log \sum_{z \in Z} p(x|z)p(z)$
  - all docs Z
  - uniform $p(z)$
  - BART $p(x \mid z)$
  - Inference via Bayes' rule: $\text{argmax}_z\, p(x_{1:t}|z)$ where t is the index of the first agent action.
- Approximate marginalization with $q(z \mid x)$: $log \sum_{z \in Z'} p(x|z)p(z) - KL[\tilde{p}(z \mid x)||\tilde{q}(z \mid x)]$
  -

7.4 Results

See table 1.

- Full marg does better than lexical baseline
- Full marg does worse than approximate marg over Z*

This is surprising, since the training setup (all Z) is closer to the testing setup (all Z), as $Z^* \subset Z$.

Two possible causes

1. Different hyperparameters: I had to use no batching for full marg, but didn't sweep over hyperparams. Learning rate should scale with batch size (citation: https://arxiv.org/abs/1706.02677)
2. There are reasonable negative documents in $Z \setminus Z^*$ that have $p(x|z) > p(x|z^*)$

7.5 Immediate next steps

- Error analysis, comparing the things full marg got wrong but approx marg got right.
- Are there negatives that were excluded by Z*, that end up making performance worse? Hyperparam sweep for full marg.
- Speed up full marg with an inference network q(z|x) in the VAE setting. There is only enough memory on the A100s to run full marg unbatched. This slows down iteration speed and will make further modeling difficult.

7.6 Sasha questions

- Isn't your model p(x, z)
  - Yes, the model is $p(x \mid z)p(z)$, with $p(z)$ uniform.
- I don't really like this experiment, because it seems to test two different things: 1) keeping the z^* in the true set, 2) approximating the marginalization. A clean experiment would be Full Marginalization vs. Approx Marginalization during training. The one that keeps around Z^* is a skyline at best, and maybe at worst not informative.
  - The approximate marginalization with Z^* will not be included in the final results, but was useful for debugging full marg and will be useful for debugging the VAE setting.
  - That said, this is a clean experiment. Only one thing is changed: the set of negatives. Approx marg w/ Z^* uses z^* and some negatives, while full marg uses z^* and all negatives. Full marg vs VAE approx marg would change both the negatives as well as whether z^* is guaranteed to be present.
- I would like your conclusions to be a little bit more clear about things like speed and methods. Is Full Marg reasonable or not?
  - Speed: Full marg takes between 5-10 hours to reach peak validation document

accuracy This is reasonable for this setting, but will become a limitation in models that must perform both sentence and document marginalization.

  - General resaonableness: Full marg is reasonable as long as it fits within memory constraints. It is reasonable for this dataset, but may not be for the other datasets.

- The name "Approx Marg" does not really make sense here, as again approx would be a version of this with the $Z*$

  - Approximate marginalization with Z^* describes the setting $\log \sum_{z \in Z^*} p(x,z), Z^* \subset Z$. Marginalization over Z is approximated over the restriction Z^*. I believe this is a precise description without jargon.

- You are much too early to worry about hyperparams, that discussion should not even be here yet.

  - I managed to get accuracy up a few points, but nothing major. Other learning rate settings resulted in very poor performance for this experiment, as fine-tuning is sensitive to hyperparameters.

- I don't really get this line "This is surprising, since the training setup (all Z) is closer to the testing setup (all Z), as Z* is a strict subset of Z". This doesn't seem surprising to me?

  - It is hard to predict whether approximate marginalization with Z^* vs full marginalization with Z would yield a better model.
  - $p(x|z)$ will learn to prefer z^* if $p(x|z^*)$ is better than other $p(x|z)$, since the gradient of the log marginal likelihood objective is the posterior $p(z|x)$ and the model has a uniform $p(z)$.
  - When would approx marg w/ Z* do better? If Z^* contains hard negatives z with $p(x|z^*) > p(x|z)$ but not negatives $p(x|z^*) < p(x|z)$, so that the model doesnt learn to prefer those hard negatives over the true z^*. This seems to be the case here.
  - When would approx marg w/ Z^* do worse? If Z^* misses some hard negatives with $p(x|z^*) > p(x|z)$.

- please provide a section in these documents with "parameterization". Is $p(z)$ parameterized?

  - $p(z)$ is uniform and therefore has no learnable parameters.

model	N	doc acc
Skyline supervised $p(z\|x)$	All	90.65
Baseline lexical	0	34.06
$p(x,z)$	0	74.2
$p(x,z)$	50	-
$p(x,z)$	100	-

Table 2: Results for document classification with a generative model. N is the number of labeled examples seen during training.

8 Experimental setup

9 Results

References

Derek Chen, Howard Chen, Yi Yang, Alex Lin, and Zhou Yu. 2021. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. *CoRR*, abs/2104.00783.