Unstructured Knowledge Access in Task-oriented Dialog Modeling using Language Inference, Knowledge Retrieval and Knowledge-Integrative Response Generation

Mudit Chaudhary[1]*, Borislav Dzodzo[1], Sida Huang[1], Chun Hei Lo[1], Mingzhi Lyu[1], Lun Yiu Nie[1], Jinbo Xing[1], Tianhua Zhang[2], Xiaoying Zhang[1], Jingyan Zhou[1], Hong Cheng[1,2], Wai Lam[1,2], Helen Meng[1,2]

[1]*The Chinese University of Hong Kong,* [2]*Centre for Perceptual and Interactive Intelligence*

muditchaudhary@cuhk.edu.hk, {bdzodzo, chlo, zhangxy, jyzhou, hcheng, wlam, hmmeng}@se.cuhk.edu.hk {sdhuang, lynie, thzhang}@link.cuhk.edu.hk, {mzlyu, jbxing}@cse.cuhk.edu.hk

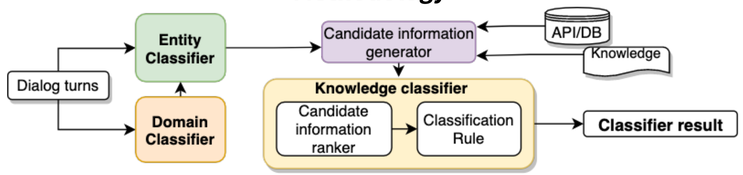
Introduction

- Dialog systems enriched with external knowledge can handle user queries that are outside the scope of the supporting databases/APIs.
- We follow the baseline provided in DSTC9 Track 1 and propose three subsystems, KDEAK, KnowleDgEFactor, and Ens-GPT, which form the pipeline for a task-oriented dialog system capable of accessing unstructured knowledge in form of FAQs.

Task 1 Knowledge-seeking Turn Detection

We propose KDEAK system to detect if the user query can be fulfilled using database or requires external knowledge from the knowledge-snippets.

Methodology



- **Domain Classifier:** Models classification as a 'Natural Language Inference (NLI)' problem to detect the domain of the current query. The query is used as the premise and the domains is used as the hypotheses. The domain with the highest entailment probability is selected.
- **Entity Classifier:** It uses a Surface Matching algorithm with heuristics to determine the query's relevant entity. It uses the domain information from previous step to reduce entity search space.
- **Candidate Information Generator:** Consolidates candidate information snippets from the database and knowledge for the selected entity into a candidate pool
- **Knowledge Classifier:** Uses NLI to find the most relevant answer to the user's query from the candidate pool. The query is classified as *True* if answering it requires external knowledge. The final classification rule is as follows:

$$label = \begin{cases} False, & if\ answer_{selected} \in Database \\ True, & otherwise \end{cases}$$

Results

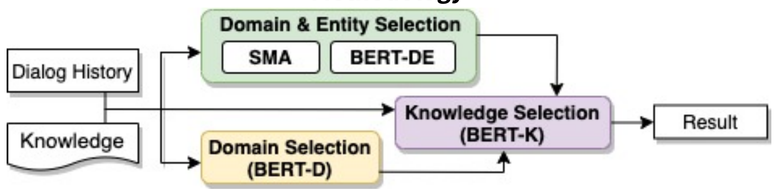
Model	Precision		Recall		F1-Score	
	VAL	TEST	VAL	TEST	VAL	TEST
GPT-2 Baseline	0.999	0.993	0.982	0.892	0.991	0.940
KDEAK (Submitted)	0.980	0.924	0.993	0.849	0.986	0.914
KDEAK (Improved)	0.993	0.985	0.986	0.952	0.989	0.968

*All authors have contributed equally

Task 2 Knowledge Selection

We propose KnowleDgEFactor to select relevant knowledge snippets for knowledge-seeking turns.

Methodology



A Factorized Approach - KnowleDgEFactor. We formulate Task 2 as a knowledge/document retrieval problem (eq. 1). We first recognize the possible target domains and entities and estimate the relevance of the domains to the dialog history before choosing knowledge snippets to narrow the search space. Factorization reduces the problem into three sub-tasks, with each module trained for target discrimination. (eq. 2)

- **Domain & Entity Selection:** Surface Matching algorithm (SMA) refined by a BERT-based domain-entity classifier (BERT-DE) is used to match possible domains and entities.
- **Domain Probability Estimation *P(d|U)*:** A multi-class domain classifier BERT-D is used to estimate the relevance of a dialog history to each domain.
- **Knowledge Probability Estimation *P(k|d,U):*** It uses a knowledge classifier, BERT-K, to estimate the relevance of a dialog history to each knowledge snippet under the selected domains and entities in Domain & Entity Selection Module.

$$\arg\max_{k_i} f(k_i \mid U_t) = \arg\max_{k_i \in_d d_i} P(d_i, k_i \mid U_t) \quad (1)$$

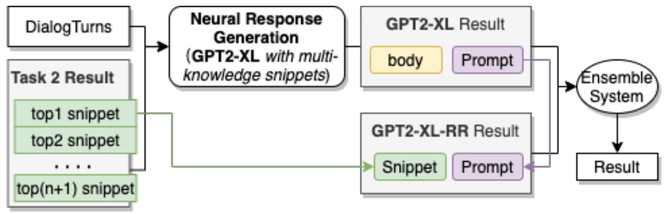
$$\arg\max_{k_i \in_d \{d_i : d_i \in D'\}} P(d_i \mid U_t) P(k_i \mid d_i, U_t) \quad (2)$$

Results

Model	Data	Task1 Source	MRR@5	R@1	R@5
Baseline	Val	Ground Truth	0.830	0.731	0.957
KnowleDgEFactor	Val	Ground Truth	**0.973**	**0.964**	**0.984**
Baseline	Test	Task1 Baseline	0.726	0.620	0.877
KnowleDgEFactor	Test	Task1 Model	**0.853**	**0.827**	**0.896**
KnowleDgEFactor	Test	Ground Truth	**0.903**	**0.867**	**0.960**

Task 3 Knowledge-grounded Response Generation

We develop an ensemble system Ens-GPT that incorporates two approaches to deal with the two scenarios that if the knowledge snippet was seen during the training of the models or not(i.e., in-domain(ID) or out-of-domain(OOD)).



Methodology

For ID cases with available training data, we adopt a **Neural Response Generation** approach. For OOD cases, we adopt a retrieval-based approach referred to as **Neural-Enhanced Response Reconstruction**. To utilize the two approaches, a decision tree is designed for the **ensemble system** *Ens-GPT*.

- **Neural Response Generation:** *GPT2-XL with multi-knowledge snippets* (*GPT2-XL* for short) leverages the large pre-trained language model GPT2-XLarge and uses multiple knowledge snippets in the input.
- **Neural-Enhanced Response Reconstruction:** *GPT2-XL Response Reconstruction* (*GPT2-XL-RR*) method forms an informative and accurate response by replacing the body of the neural generated response with the top-ranking snippet, while preserving the prompt in the generated response.
- **Ensemble System:** *Ens-GPT* first checks if the user query is ID or OOD, which is indicated by the domain of the top-ranking retrieved snippet. If the query is OOD, the confidence value p of the top-ranking snippet will be checked. If it is high enough, GPT2-XL-RR is used for response generation. Otherwise, the ensemble method chooses GPT2-XL.

Results

Model	BLEU-1	BLEU-4	METOR	ROUGE-1
DSTC9 Track 1 Baseline	0.3031	0.0655	0.2983	0.3386
GPT2-XL	0.3550	**0.1048**	0.3593	0.3972
GPT2-XL-RR	0.3521	0.1042	**0.3780**	0.3957
Ens-GPT	**0.3550**	0.1040	0.3594	**0.3976**

Conclusion

- The improved KDEAK model outperforms the baseline and is robust to unseen domains
- We formulate Task 2 as a knowledge retrieval problem, factorize it into 3 subproblems and resort to a 3-module KnowleDgEFactor approach
- Ens-GPT integrates multiple retrieved knowledge-snippets to enrich knowledge and improve robustness. The domain and retrieved snippets logits are used to build the ensemble system