Meta-Path Augmented Response Generation

Yanran Li, Wenjie Li
Department of Computing, The Hong Kong Polytechnic University
{csyli,cswjli}@comp.polyu.edu.hk

Abstract

We propose a chatbot, namely MOCHA to make good use of relevant entities when generating responses. Augmented with meta-path information, MOCHA is able to mention proper entities following the conversation flow.

Introduction

Recent years have witnessed a rise in building automatic chit-chat conversational systems (a.k.a. chatbots). To generate informative responses, these chatbots need to be aware of conversation-related knowledge. For example, when talking about a film, it is natural to mention its director and actors.

We thus endow the chatbots with the entity reasoning mechanism, i.e., the ability to mention proper entities with reference to an associated knowledge base (KB) when generating responses. To ensure response quality, the generated entities should not only be relevant, but also coherent to the conversation flow. When previous conversation had focused on the actors of a film, it would be weird to suddenly mention the film's writer. Inspired by (Dong, Chawla, and Swami), we capture conversation flow using meta-paths over the mentioned entities. A meta-path is a sequence of object types modeling a particular semantic relationship. An example meta-path of entity mentions is *actor→film→film*. Since meta-path information greatly reflects the conversation flow, it is better to mention entities following the meta-path when generating responses.

In this work, we develop a **M**eta-path augmented Kn**O**wledge-grounded **CHA**tbot, namely MOCHA. Given a conversation, our MOCHA firstly collects relevant entities as candidates to benefit response generation. Then, MOCHA captures the conversation context and generates responses based on an devised encoder-decoder architecture. The encoder compresses the input utterance(s) into a context vector. The decoder is augmented with a pointer gate (Vinyals, Fortunato, and Jaitly 2015) to decide when to mention an entity and conduct entity reasoning over the pre-collected candidate entities based on meta-path information. Particularly, we define 10 most popular meta-paths observed in our conversation data, and encode them into vectors using metapath2vec approach (Dong, Chawla, and Swami). Finally, MOCHA is able to generate entities by firstly comparing the context representation with each of the meta-path vectors, and then attending on the candidate entities that follow the most similar meta-path. On the movie corpus we build, our MOCHA significantly outperforms the compared models.

Meta-path Embedding

Given the entity types from $\mathcal{A}$, a meta-path $\mathcal{P}$ is denoted as $A_1 \rightarrow A_2 \rightarrow \ldots \rightarrow A_{L+1}$, which defines a semantic relation between types A_1 and A_{L+1}, where L is the path length.

To augment response generation, we summarize 10 popular meta-paths according to the statistics of the conversation corpus. Each meta-path connects 3 entity types (length of 2), formed by entity mentions in their original order from a same conversation. We depict these 10 meta-paths in the Supplementary Material (Suppl. in short).[1]

After defining these meta-paths, we generate path instances and embed each 2-length instances using a GRU following (Dong, Chawla, and Swami). Since a meta-path can produce multiple instances, we aggregate all the instance embeddings into a single vector using a max-pooling operation. As a result, for each of 10 meta-paths, we obtain a vector representation $\mathbf{p}_m$, where $\forall m \in \{1, \ldots, 10\}$.

Model: Mocha

Formally, a conversation consists of utterance sequences $\mathbf{x} = \{\mathbf{u}^1, \ldots, \mathbf{u}^T\}$, where T is the turn number. Each utterance is a sequence of words. Hence, the input of chatbot is a word sequence $\mathbf{x} = \{x_1, \ldots, x_{N_x}\}$, and the output is a response $\mathbf{y} = \{y_1, \ldots, y_{N_y}\}$, where N_x and N_y are the token numbers. MOCHA is a knowledge-grounded chatbot, consisting of three main components, as follows:

Entity Collector. MOCHA firstly shortlists a set of conversation-related entities E from KB, which is detailed described in Suppl. For decoder use, we employ TransE (Bordes et al. 2013) to transform the entities into dense embeddings as $\mathbf{E} = \{\mathbf{e}_n\}$, where $\forall n \in \{1, \ldots, N_e\}$.

Context Encoder. We then embed the input utterances $\mathbf{x}$ using a bi-directional GRU.[2] The resulting representation at each time step is the concatenation of each direction's state, i.e., $\mathbf{h}_t = [\overleftarrow{\mathbf{h}}_t, \overrightarrow{\mathbf{h}}_t]$, which is then fed to the decoder.

[1] yanran.li/mocha
[2] We did not see clear improvements using hierarchical encoder.

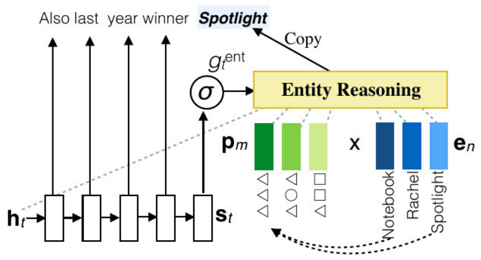

Figure 1: Entity-aware Decoder. Best viewed in color.

Entity-aware Decoder. As shown in Figure 1, we augment the GRU-based decoder with a gating variable g_t^{ent} (Vinyals, Fortunato, and Jaitly 2015) that decides whether to generate an entity using p^{ent} or to omit a general word using p^{gru}. The gate g_t^{ent} is trained on:

$$g_t^{\text{ent}} = \sigma(\mathbf{W}_g \mathbf{s}_t)$$

When the gate is "open", the decoder conducts entity reasoning by taking into account the meta-path information. It firstly approximates how close each meta-path $\mathbf{p}_m$ is to the context $\mathbf{h}_t$, and obtain the attention weights $\boldsymbol{\alpha}_t$ as:

$$\boldsymbol{\alpha}_t \sim \exp(\mathbf{P}\mathbf{W}_p \mathbf{h}_t)$$

where $\mathbf{P}$ is the matrix consisting of the mata-path vectors $\mathbf{p}_m$. Then, we apply another attention mechanism on $\mathbf{e}_n$ and obtain the corresponding weights $\boldsymbol{\beta}_t$. Intuitively, the generated entity should belong to the ending type (A_3) in the attended meta-paths. To do so, we align the entity weights with their corresponding path weights, and multiply the two weights as the output probability. Finally, the decoder generates a candidate entity by:

$$p^{\text{ent}}(y_t|\mathbf{h}_t, \mathbf{P}, \mathbf{E}) = \begin{cases} \alpha_{ti}\beta_{tj}, & \text{if } y_t = e_j \text{ and } e_j \mapsto A_i \\ 0, & \text{otherwise} \end{cases}$$

When referreing is needed, the decoder directly copies the entity with the highest probability. In this way, the generated response is expected to follow the conversation flow by approximating the context representation $\mathbf{h}_t$ with both meta-path information and candidate entities.

Experiments

To validate, we build a movie conversation corpus consisting of roughly 9,000 conversations. We also build a movie KB to link with. The corpus statistics and KB schema are in Suppl.

Compared Models: (1) ATTN: a attention-based Seq2Seq model. (2) HRED (Serban et al. 2016): a hierarchical Seq2Seq approach. (3) FACT (Ghazvininejad et al. 2017): a knowledge-grounded model that consumes textual, unstructured facts. (4) GENDS (Zhu et al. 2017): its decoder is also augmented with entity reasoning, but without meta-path taken into account.

Evaluation Metrics: (1) BLEU-3; (2) Dist-1 (Li et al. 2016) calculates the ratios of unigrams; (3)(4) Precision and Recall (Zhu et al. 2017) examines the overlapping on entity mentions in the generated responses.

Model	KB	BLEU-3	Dist-1	Prec.	Recall
ATTN	✗	0.18	0.03	0.14	0.14
HRED	✗	0.17	0.02	0.13	0.14
FACT	✓	0.06	0.16	0.13	0.08
GENDS	✓	0.68	0.16	0.37	0.38
MOCHA	✓	**0.82**	**0.19**	**0.48**	**0.51**

Table 1: Experimental Results.

We constrain the vocabulary to 25,000 words. The embeddings size are 300 and the hidden state vectors are 512. Models are trained using Adam optimizer.

As shown in Table 1, ATTN and HRED perform the worst because they are the models that have no access to external KB. We then compare the other three models to find which one(s) utilize(s) knowledge more effectively. Obviously, FACT lags far because it utilizes knowledge described in unstructured text, i.e., *Titanic stars Leonardo as...*. Its disappointing performance suggest that it is more effective to inject structural knowledge into Seq2Seq models.

Remarkably, MOCHA and GENDS are the best and second best models. They are the only models with explicit entity reasoning mechanism to generate entities from shortlisted candidates. This proves the necessity of such mechanism for the chatbots. Different from GENDS, MOCHA attends on entities based on meta-path information, and prefers those entities that follow the most similar meta-path to the current conversation context. Considering the attention weights on the meta-path vectors, MOCHA reduces possibilities of generating incoherent entities when they diverge from the conversation flow. Drawing on the highest scores achieved by MOCHA, it is beneficial to augment response generation using meta-path information.

Acknowledgements

The work described in this paper was supported by a HK GRF grant (No. PolyU 152040/18E).

References

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *NIPS*.

Dong, Y.; Chawla, N. V.; and Swami, A. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*.

Ghazvininejad, M.; Brockett, C.; Chang, M.-W.; Dolan, B.; Gao, J.; Yih, W.-t.; and Galley, M. 2017. A knowledge-grounded neural conversation model. *arXiv preprint arXiv:1702.01932*.

Li, J.; Galley, M.; Brockett, C.; Gao, J.; and Dolan, B. 2016. A diversity-promoting objective function for neural conversation models. In *EMNLP*.

Serban, I.; Sordoni, A.; Bengio, Y.; Courville, A. C.; and Pineau, J. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. In *NIPS*.

Zhu, W.; Mo, K.; Zhang, Y.; Zhu, Z.; Peng, X.; and Yang, Q. 2017. Flexible end-to-end dialogue system for knowledge grounded conversation. *arXiv preprint arXiv:1709.04264*.