Goal-oriented Knowledge-driven Neural Dialog Generation System

Ao Luo, Chen Su, and Shengfeng Pan

Zhuiyi Technology, Shenzhen
{frankluo, chensu, nickpan}@wezhuiyi.com

Abstract. In this paper, we propose a goal-oriented knowledge-driven neural dialog generation system, which leads the conversation based on a knowledge graph. During the conversation, the system has to actively integrate appropriate knowledge conditioned on current dialog state, and then generate coherent, fluent and meaningful responses. We use ERNIE as our backbone model, proposing a fine-tuning scheme to first pre-train on knowledge graph and dialog sequence, and then fine-tune to generate the next response. We extend multi-task learning in multi-turn dialog generation to improve consistency. We show that with well-designed transfer learning, ERNIE shows competitive performance on a knowledge-grounded dialog generation task. In the Baidu knowledge-driven dialog competition, our best single model achieved 4th in the automatic evaluation stage with 47.03 f1 score and 0.417/0.281 BLEU1/BLEU2 score, and ranked 1st in the final human evaluation stage, with descent topic completion performance(1.81/3) and highest coherence score(2.59/3).

Keywords: knowledge-grounded dialog generation · transfer learning

1 Introduction

Non-goal-oriented multi-turn dialog systems usually suffer from inconsistency [7] and the tendency to produce non-specific meaningless answers [6]. Recently, several datasets and models are proposed to integrate personality [3, 18] or external knowledge [4, 8] into dialog generation, to produce the response with richer meaning and better consistency. Based on that, Baidu steps forward and releases a dataset DuConv [16], encouraging models to proactively plan over the knowledge graph and lead in a conversation(introduce a new topic or maintain current topic), instead of only generating responses to answer questions.

Transfer learning has gained huge success in many tasks of natural language processing, thanks to the pre-trained language models [10, 11, 2, 1, 12], which learn rich deep contexture representations in the pre-training phase that could be transferred to downstream tasks. There are two existing strategies for applying pre-trained language representations to down-stream tasks: feature-based and fine-tuning, whose performance depends on the similarity of pretraining and target tasks [9].

We propose to apply transfer learning on the latest knowledge-based dialog generation datasets. Recently, a fine-tuned OpenAI GPT model TransferTransfo [15] shows strong improvements over the current state-of-the-art like memory augmented seq2seq and information-retrieval models [18] on the PERSONA-CHAT dataset, which shows the advantages of combining transfer learning with OpenAI GPT on personality-based dialog system. However, TransferTransfo might fail to generalize to a knowledge-grounded dataset like DuConv, since

1. GPT is a unidirectional auto-regressive language model. It could encode consecutive sequences such as persona in the PERSONA-CHAT dataset, but it might not make much sense to encode a knowledge graph in the DuConv, where a bidirectional encoder might be superior.
2. GPT's attention mask on the pre-training phase constraints the dependency of a sequence, which leaves us no space to optimize the structure of the input sequence.

BERT [2], however, using a bidirectional transformer encoder, could encode more complex dependency and is better in semantic representation. Thus, we want to investigate the performance of BERT in the knowledge-grounded dataset. In this work, we propose a simple variant of the ERNIE model [13], which is a Chinese version of BERT with knowledge masking training strategy, and propose a corresponding adaptation scheme, on goal-oriented knowledge-grounded dialog generation task. Our fine-tuned model shows competitive performance in both goal completion and coherence.

2 Task Data and Evaluation

Baidu's knowledge-driven dialogue competition aims to investigate machines' ability to conduct human-like conversations, in a proactive way [16]. Generally, given a set of topic-related background knowledge and dialog history, the model is expected to generate the next response which keeps the conversation coherent and informative under the guidance of the provided goal. Most importantly, the model is required to proactively shift topic from one to another in the conversation. The dataset includes 30k sessions, about 120k dialogue turns, of which 100k are training set, 10k are development set and 10k are test set.

We describe one session(Fig 1) in detail. A goal is represented as a path {start, topic_a, topic_b} plus key spo(s), such as {topic_a, relation, topic_b} or a pair of {topic_a, property, value} and {topic_b, property, value}, which connect the two topics. It instructs the model to first introduce topic_a, and then shift to topic_b using the key spo(s). The background knowledge is organized in the form of {entity, property, value}, where entity here is either topic_a or topic_b. The number of background knowledge for each topic is usually 7 or 8, including both factoid knowledge and non-factoid knowledge such as comments. Agent is assumed to speak first, generating response on $h_0, h_2, h_4...$; person is assumed to follow agent, generating response on $h_1, h_3, h_5....$

```
{"goal":
  [["START", "阳光灿烂的日子", "王朔"],
  ["王朔", "代表作", "阳光灿烂的日子"]],
 "knowledge": [["阳光灿烂的日子", "时光网 短评", "70 年代 少年 人 的 成长 经历 ， 太 过 真实 ， 再回首 至于 刺眼 的 日光 灼 目"],
  ["阳光灿烂的日子", "主演", "宁静"],
  ["阳光灿烂的日子", "上映 时间", "1994年9月9日"],
  ["阳光灿烂的日子", "类型", "剧情"],
  ["阳光灿烂的日子", "领域", "电影"],
  ["王朔", "评论", "才华横溢 ！ "],
  ["王朔", "毕业 院校", "北京四十四中学"],
  ["王朔", "主要 成就", "第53届洛迦诺国际电影节 主 竞赛 单元 – 金豹奖"],
  ["王朔", "性别", "男"],
  ["王朔", "职业", "编剧"],
  ["王朔", "领域", "明星"],
  ["阳光灿烂的日子", "是否 上映", "已 上映"],
  ["阳光灿烂的日子", "时光网 短评", "有点 西西里 的 感觉 。"],
  ["阳光灿烂的日子", "时光网 评分", "8.5"],
  ["阳光灿烂的日子", "导演", "姜文"]],
 "conversation":
  ["我 发现 姜文 的 电影 产量 不 高 ， 但是 质量 都 挺 高 的 。",
   "同感 ， 那 你 觉得 你 印象 最深 的 一部 姜文 的 作品 是 什么 ？ ",
   "阳光灿烂的日子 吧 ， 有点 西西里 的 感觉 。",
   "我 也 觉得 这部 电影 不错 ！ ",
   "嗯 呀 ， 它 是 一个 年代 的 缩影 吧 。",
   "对 呀 ， 可能 姜文 只是 把 他 自己 经历 的 给 拍 了 出来 吧 。",
   "但是 里面 那位 主演 真 的 是 才华横溢 。",
   "你 说 的 是 哪 一位 ？ "
   "王朔 啊 ， 是 北京四十四中学 毕业 的 那位 。"] }
```


Fig. 1. Data

The automatic evaluation involves three metrics, including 1) char-based F1-score of output responses against golden responses; 2) word-based precision of output responses against golden responses; 3) diversity of the output responses. The F1 and BLEU are main metrics and DISTINCT is the auxiliary metric.

Human evaluation is based on criteria of coherence and goal completion. The coherence measures the overall fluency of the whole dialog and the goal completion measures how good the given competition goal is finished. The details could be found in [16]. This is consistent with the current main dialog challenges [3].

3 Model

Our model is adapted from Baidu ERNIE [13], which pre-train BERT base (12 layers, 12 attention heads, 768 hidden dimensions) on mixed corpus Chinese Wikipedia, Baidu Baike, Baidu news, and Baidu Tieba, using entity-level and phrase-level masking. In the DuConv task, we did the following modification:

Input Similar to TransferTransfo [15], we prefer to concatenate dialog history and the next response as a sequence; conditioned on the dialog history, model extends the history to generate the next response, like what GPT-2 does in generating conditional synthetic text [12]. In training, we use target dialog history; during inference, we use the concatenated sequence of generated response and person's response as dialog history. As in Fig 2, we also concatenate goal and the knowledge as a sequence, appending in the front of the conversation. Notice, we always use goal and full knowledge for every agent response generation in a

dialog session. We expand the segment types of ERNIE, let segments as tags to differentiate each component in a knowledge conversation sequence. We finally acquire a token sequence and a corresponding segment type sequence. The input embedding is the summation of word embedding, segment embedding, and positional embedding.

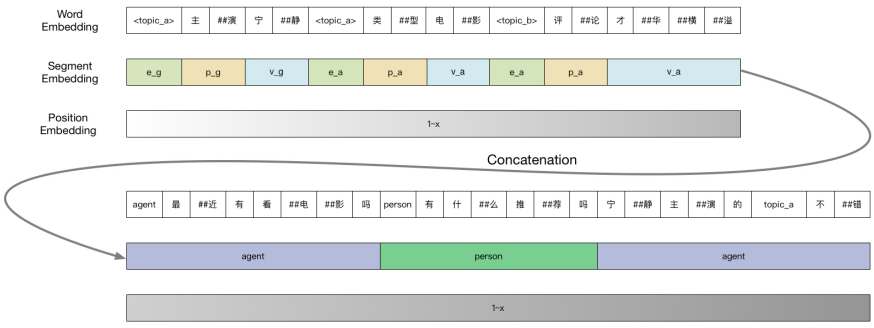

Fig. 2. Input embedding ($E_{input} = E_{word} + E_{segment} + E_{position}$)

Attention Mask Similar to transformer decoder [14, 11], we apply a future mask on conversation sequence, to allow token in conversation sequence only attend on previous tokens in self-attention. In addition, we adjust the attention mask on goal and knowledge sequence, to mimic the structure of the knowledge graph, as illustrated in Fig 3. This new attention mask maintains bidirectional attention inside each knowledge and mutual attention between directly connected knowledge, in the meantime time, stops the mutual attention between knowledge that don't directly connect. A goal is allowed to connect with knowledge.

Primal and A variant In primal setting (left in Fig 4), a single model is used to encode a knowledge sequence and decode history and next response. In a variant setting (right in Fig 4), we consider using different weights on knowledge sequence and conversation sequences, since the difference between knowledge sequence and conversation in structure, but we maintain the model architecture unchanged. It's basically equal to that a ERNIE base as an encoder to encode knowledge graph, and another ERNIE base as a conversation decoder to decode the next response; they are connected through attention mechanism in the 12 layers of transformer, as in the equation 1.

$$\begin{aligned} q^l &= W_q(h_{dialog}) \\ k^l &= [k_{knowledge}; W_k(h_{dialog})] \\ v^l &= [v_{knowledge}; W_v(h_{dialog})] \\ h^{l+1} &= MultiheadAttention(k^l, v^l, q^l) \end{aligned} \tag{1}$$

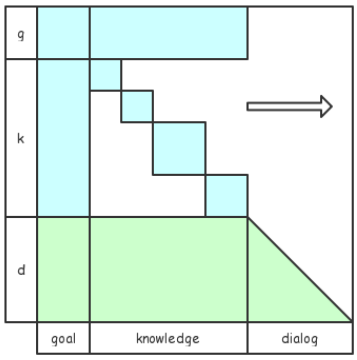

Fig. 3. Attention mask

Notice [*; *] means concatenation on the sequence length dimension.

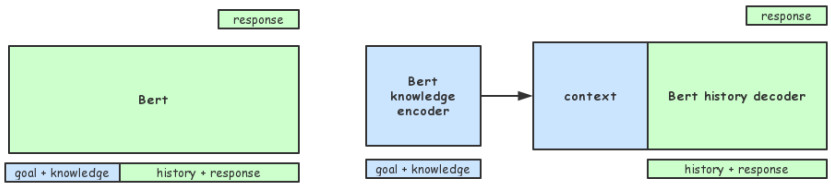

Fig. 4. Model Primal & A variant left is primal, which is a single ERNIE; right is ERNIE variant, which is still ERNIE but with different weights on knowledge and conversation

Generation We pick the last token in the final layer of ERNIE, and project into vocab size dimension using a linear projection, and finally a log-softmax layer. The next prediction is the token that maximizes the log probability. This linear projection shares the same weights as word embedding.

4 Training

4.1 Input Features

We describe our inputs from Word, Segment type, and Concatenation.

Word We use generalization token topic_a, topic_b to replace the original entity in knowledge and dialog, and replace back in the post-processing. In the DuConv dataset, words are segmented in both knowledge and conversation, and the BLEU metrics are based on the word level, probably because of its downstream application. To retain the word segmentation, we use WordPiece segmentation from ERNIE and first split by space. Specifically, we utilize the subword starting with "##" in ERNIE, to represent the current character is part of the front word. For example, "ABCDEF GH E" will be split into "A", "##B", "##C", "##D", "##E", "F", "G", "##H", "E". In the post-processing, the subword starting with "##" will be merged into the front word.

Segment type We tag each token in an input sequence using a set of predefined segment types, based on which component it belongs to in a dialog session. The segment helps the model reconstruct the knowledge graph and conversation flow in a concatenated sequence. The basic components in goal and knowledge are {entity, property, value}, and we want to differentiate them in goal and knowledge from each topic. Thus the segment types for goal and knowledges are {*entity_goal, property_goal, value_goal*} and {*entity_a, property_a, value_a, entity_b, property_b, value_b*}. We decompose the conversation into two types {*agent, person*}, according to whether this response comes from agent or person; we could also decompose the conversation based on the absolute position in a conversation, $\{h_0, h_1, h_2..., h_max\}$, where h_i is the ith response in the whole conversation. In a goal-oriented multi-turn dialog generation task, the model is required to complete the goal in a limited number of turns. Thus, the position signal could be used as an auxiliary signal to dialog state. In the general case, we could sum these two sets of segments embeddings.

Concatenation Intuitively, knowledge is positional invariant in a concatenated sequence, but positional variant within itself. Thus, we shuffle the knowledge for each topic in each time, and then concatenate them together. This also improves the generalization of the model. As we describe above, goal, knowledge, and conversation are concatenated as {*goal, knowledge_a, knowledge_b, dialog history, next response*}.

4.2 Pre-training

We initialized the model from ERNIE, and first pre-train on knowledge, and then pre-train on whole conversation, finally finetune on response from agent. We first introduce our pre-training scheme.

Knowledge Pre-training Since the knowledge sequence, concatenated from the knowledge graph, is different from the original corpus ERNIE pre-trained on, we first pre-train our model on knowledge sequences using masked LM task similar to ERNIE. The objective is equivalent to the equation 2. Based on the structure of the knowledge graph, we do not mask entity token since it's too easy; also, for each knowledge, we do not mask property and value at the same time because it's too difficult. This process helps the model adapt to the knowledge sequence.

$$\mathcal{L}_{knowledge\_pre-train} = -\log p_\theta(\bar{x}|\hat{x}) \approx -\sum_{i=1}^{|\bar{x}|} \log p_\theta(\bar{x}_i|\hat{x}) \tag{2}$$

where $\bar{x}$ represents the tokens masked, where $\hat{x}$ represents the tokens are not masked.

Sequence Pre-training To fully use the conversation data and improve the efficiency of training, we concatenate the whole conversation into one sequence, and apply auto-regressive sequence loss on the concatenated conversation, including the response from agent and person, which shortens the training time. In addition, we randomly choose two question-answer pairs in the conversation and exchange their position in 50 % of the time, and append a [CLS] token at the end of the conversation. The final hidden vector in the [CLS] position is chosen and passes through a linear classifier to do a binary classification, to differentiate whether there is a position exchange. The final loss is the weighted sum of sequence loss and classification loss, as in equation 3, and α is a hyperparameter. We propose it helps 1) let model focused on the last question, improving generalization 2) classification helps model figure out the inconsistency in multi-turn conversation, and then generate response more logic reasonable. Notice, when doing the exchange, we only exchange tokens and segment types are decided by the position after exchange. In this task, we only exchange the pair, in which the question is from person and the answer is from agent.

$$\begin{aligned} \mathcal{L}_{sequence\_pre-train} &= \alpha\mathcal{L}_{seq} + \mathcal{L}_{cls} \\ \mathcal{L}_{seq} &= -\sum_{i=0}^{|\mathbf{H}|}\sum_{t=0}^{|\mathbf{h}_i|} \log p_\theta(x_t|goal, knowledge, \mathbf{h}_{<i}, \mathbf{x}_{<t}) \end{aligned} \tag{3}$$

where $\mathbf{h}_{<i}$ represents dialog history before current response $\mathbf{h}_i$, and $\mathbf{x}_{<j}$ represents previous generated tokens in current response $\mathbf{h}_i$.

4.3 Finetuning

In this phase, we train the model to generate the next response from agent, which is our final objective, with a classification task to classify the next sentence. We only concatenate history and the next response from the agent and apply NLL loss on the response, instead of the whole conversation as in sequence loss. The final loss is the weighted summation of response loss and classification loss, as in equation 4, and β is a hyperparameter. We extend the next sentence classification task in the original BERT to a multi-turn dialog setting. Instead of only predicting whether the next sentence is the ground truth sentence, we do a 3-way classification on 1) ground truth 2) random sampled from the large dialog corpus 3) random sampled from the current session except for the ground truth. Binary classification is too simple for a knowledge-grounded task since any response carrying knowledge different from the current dialog session will be wrong in very high probability. Therefore, the third class we added, could

force the model to learn what is the appropriate sentence considering the dialog history and the last question.

$$\begin{aligned} \mathcal{L}_{finetune} &= \beta \mathcal{L}_{response} + \mathcal{L}_{cls} \\ \mathcal{L}_{response} &= -\sum_{j=0}^{|\mathbf{h}_i|} \log p_\theta(x_j | goal, knowledge, \mathbf{h}_{<i}, \mathbf{x}_{<j}) \end{aligned} \tag{4}$$

where $\mathbf{h}_i$ is the current generated response, and $\mathbf{x}_{<j}$ represents the previous generated tokens in current response $\mathbf{h}_i$.

4.4 Decoding

We utilize a very simple decoding strategy. To improve the diversity, we split the beam size into several groups; in each group, we do a normal beam search with length penalty, with the beam size as $beam_size/groups$. We use rules to improve choosing strategy: 1) penalize the candidates that include topic b while topic a not in the dialog history 2) penalize the sentences that include topic a, while the topic a already in the dialog history.

4.5 Ensemble

Here, we describe a very simple ensemble strategy. When we do the next sentence classification in response finetuning, we get a next sentence classifier. We use the label score from this classifier to choose final next response from several models.

4.6 Implement details

We train the model in the order described above. We don't use other corpus in this competition, but we are exploring that and find it beneficial. In the variant model we mentioned in section 3, we experiment using the weights from primal model to initialize the weights for both knowledge encoder and conversation decoder.

Pre-training In the knowledge pre-training, we set batch_size to be 512, learning rate to be 6.25×10^{-5}, use normal BertAdam with 10 % warmup and learning rate decreasing linearly. We train for 2 epochs until the knowledge loss drops slowly. We randomly mask 15 % words; in 80% we use [MASK] to replace masked words, in 10% we use random tokens, in 10% we use original word token. In the sequence pre-training, we use a similar learning schedule as knowledge pre-train, except we set batch size as 32, and train about 8 epochs, with scale α set as 2. It takes about 200 minutes on a 20G Nvidia Tesla P40, about 25 minutes per epoch with 625 batches. In pre-training, we only update segmentation embedding in the first few batches, and then unfreeze all weights to update.

Finetuning In the response fine-tuning, we set the batch size to be 64, learning rate 1.25×10^{-5}, same as above optimization schedule, with classification

scale β as 2. Finetuning will continue 1 epoch. This process will take about 4 hours on a 20G Nvidia Tesla P40.

Decoding In the decoding phase, since the generated tokens do not change the representations from the knowledge and previous dialog history. We could reuse the intermediate key and value results, which improves efficiency.

4.7 Other baselines

We also implement other models as a comparison:

1. GPT. Similar to our ERNIE adaptation in input embedding and pre-train scheme, except that we replace ERNIE with an OpenAI GPT which we trained mainly on Baidu Baike, and use a normal future mask [14, 11].
2. Wiki model, from Wizzard of wiki [4]. Same with the original paper, we use a two steps transformer encoder and decoder. The encoder encodes the knowledge and dialog history independently and the decoder decode the response conditioned on encoded contexts; an additional attention mechanism is used to choose the knowledge based on encoded representation.
3. DeepCopy [17]. DeepCopy is the extension of CopyNet, which can copy from multiple sources. We use stacked long-short-term-memory as encoder and decoder, enhanced by residual connection and highway layer. We initialize the LSTM layers randomly and word embedding using Glove.

5 Results

5.1 Compare to other models, GPT, WIKI, DeepCopy

Table 1 is the comparison of single models on automatic metrics on the dev set. Models with only word embedding initialized like baseline, wiki and DeepCopy perform worse than models initialized from pre-trained language models like GPT and ERNIE primal and variant. In addition, the response coherence of finetuned models are obviously better than models trained from scratch; we don't compare their scores, but the final human evaluation in the competition proves that. The primal ERNIE is competitive with GPT and the variant ERNIE slightly outperforms GPT. The advantage of ERNIE is stronger in word-level metrics BLEU1 and BLEU2, probably because ERNIE vocab could easily encode and reconstruct word segmentation, while GPT needs well-designed post-processing. Table 2 is the comparison of single models and ensemble models on automatic metrics and human metrics. Ensemble models are stronger than single models in automatic metrics. However, on human evaluation, ensemble models don't achieve better results. It could be explained by the not perfect ensemble strategy, or it's because ensemble models favorite safer and non-diverse responses. Overall, it shows that 1) pre-trained language models outperform other models in a knowledge-grounded dialog generation task, especially in the coherence from the human point of view. 2) ERNIE model with appropriate adaptation and pre-training can achieve competitive or even better results over

Table 1. single model comparison on automatic metircs. baseline from DuConv [16]; Wiki, DeepCopy and GPT are introduced in section 4.7; primal is the single ERNIE, variant is the ERNIE with different weights on knowledge and conversation, or could be viewed as two ERNIE like model for encoder and decoder respectively; variant-2 has the same structure as variant, except that we initialize the (knowledge) encoder and (conversation) decoder with weights from single ERNIE after pre-training. It is worth mention that variant is not the same as primal. Although the encoder and decoder are initialized with the same weights, they do not share the same weights during training.

Model	F1	BLEU1/BLEU2	DISTINCT 1&2
baseline	36.21	0.320/0.169	0.072/0.156
Wiki	42.17	0.365/0.241	0.087/0.230
DeepCopy	43.10	0.365/0.234	0.107/0.287
GPT	44.30	0.385/0.256	0.112/0.301
primal	44.30	0.388/0.266	0.120/0.322
variant	44.21	0.395/0.268	0.114/0.308
variant-2	**44.80**	**0.400/0.270**	0.112/0.302

GPT in the knowledge-grounded dialog generation task. 3) Single models might be better than ensemble models in human evaluation.

Table 2. comparison single model and ensemble. these human evaluation is conducted by ourself using 200 examples in dev set. Our final human evaluation in the competition is 1.81 in goal completion and 2.59 in coherence using single variant-2 model.

	Automatic		Human* (0,1,2,3)	
Model	F1/BLEU1/BLEU2	DIST1/DIST2	completion	coherence
GPT	44.30/0.385/0.256	0.112/0.301	1.71	2.63
primal	44.30/0.388/0.266	0.120/0.322	1.80	2.51
variant-2	44.80/0.400/0.270	0.112/0.302	1.82	2.62
ensemble-GPT	45.20/0.390/0.262	0.109/0.296	1.81	2.52
ensemble-primal	45.05/0.399/0.269	0.110/0.302	1.83	2.49
ensemble-variant-2	**45.25/0.403/0.271**	0.115/0.318	1.86	2.58

In this Baidu knowledge-driven dialog competition, we do not use other corpus to pre-train BERT on a generation setting, which constraints the BERT's ability to generate sentences. A very recent paper [5] proposes that using auto-regressive, partial auto-regressive and bidirectional language objectives to pre-train on BERT (unified language pretraining) could improve BERT's ability to generate, and BERT generative model achieves the state-of-the-art on CoQA dataset over other generative models by a large margin. We believe that with pre-training on other corpus, our BERT model could further improve.

5.2 Contribution of different parts.

Table 3. Ablation Analysis. we use variant-2 as the base, to modify each component independently, including remove segment embedding, remove knowledge pre-train, remove sequence pre-train, remove risk minimization, modify attention mask to simple bidirectional, and remove multi-task.

Model	F1/BLEU1/BLEU2
variant-2	**44.80/0.400/0.270**
- segment embedding	42.25/0.370/0.253
- knowledge pre-train	44.03/0.382/0.258
- sequence pre-train	42.65/0.372/0.256
+ simple attention mask	44.01/0.380/0.258
- multi-task	42.32/0.371/0.255

We further investigate our best single ERNIE model by comparing the contribution of each adaptation in table 3. It shows that segment embedding and sequence pre-train are especially important in our model. It's reasonable because 1) segment embedding helps model recognize different components of knowledge and structure of dialog history easily, 2) sequence pre-train expand the data by incorporating the responses from an agent, which improves the ERNIE's ability on generation. Knowledge pre-train and attention mask designed for knowledge graph also improves performance decently, which mainly improves the model on encoding knowledge. In addition, multi-task helps improve on both automatic metrics and human evaluation scores, which mainly helps in maintaining a better consistent conversation.

6 Conclusion

In this work, we experiment applying ERNIE in a goal-oriented knowledge-based dialog generation dataset DuConv. Although BERT outperforms other language models on natural language understanding tasks, few people use it in natural language generation tasks, since its bidirectional encoding setting. Our results prove that with proper adaptation on pre-training, modifying attention mask, and expanding segment embeddings, BERT could also shine in the knowledge-grounded dialog generation. In addition, we extend the multi-task in multi-turn dialog setting that helps improve the consistence of conversation. Our best single ERNIE base model helps us win the first place in the Baidu knowledge-driven dialog competition.

References

1. Dai, Z., Yang, Z., Yang, Y., Cohen, W.W., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)

2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
3. Dinan, E., Logacheva, V., Malykh, V., Miller, A., Shuster, K., Urbanek, J., Kiela, D., Szlam, A., Serban, I., Lowe, R., et al.: The second conversational intelligence challenge (convai2). arXiv preprint arXiv:1902.00098 (2019)
4. Dinan, E., Roller, S., Shuster, K., Fan, A., Auli, M., Weston, J.: Wizard of wikipedia: Knowledge-powered conversational agents. arXiv preprint arXiv:1811.01241 (2018)
5. Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., Hon, H.W.: Unified language model pre-training for natural language understanding and generation. arXiv preprint arXiv:1905.03197 (2019)
6. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. arXiv preprint arXiv:1510.03055 (2015)
7. Li, J., Galley, M., Brockett, C., Spithourakis, G.P., Gao, J., Dolan, B.: A persona-based neural conversation model. arXiv preprint arXiv:1603.06155 (2016)
8. Liu, S., Chen, H., Ren, Z., Feng, Y., Liu, Q., Yin, D.: Knowledge diffusion for neural dialogue generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1489–1498 (2018)
9. Peters, M., Ruder, S., Smith, N.A.: To tune or not to tune? adapting pretrained representations to diverse tasks. arXiv preprint arXiv:1903.05987 (2019)
10. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
11. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf (2018)
12. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8) (2019)
13. Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., Wu, H.: Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223 (2019)
14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
15. Wolf, T., Sanh, V., Chaumond, J., Delangue, C.: Transfertransfo: A transfer learning approach for neural network based conversational agents. arXiv preprint arXiv:1901.08149 (2019)
16. Wu, W., Guo, Z., Zhou, X., Wu, H., Zhang, X., Lian, R., Wang, H.: Proactive human-machine conversation with explicit conversation goals. arXiv preprint arXiv:1906.05572 (2019)
17. Yavuz, S., Rastogi, A., Chao, G.l., Hakkani-Tür, D., AI, A.A.: Deepcopy: Grounded response generation with hierarchical pointer networks. NIPS (2018)
18. Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., Weston, J.: Personalizing dialogue agents: I have a dog, do you have pets too? arXiv preprint arXiv:1801.07243 (2018)