Knowledge Enhanced Opinion Generation from an Attitude

Zhe Ye[1,3], Ruihua Song[2(✉)], Hao Fu[3], Pingping Lin[3], Jian-Yun Nie[4], and Fang Li[1]

[1] Shanghai Jiao Tong University, Shanghai, China
fli@sjtu.edu.cn
[2] Renmin University of China, Beijing, China
songruihua_bloon@outlook.com
[3] Microsoft, Beijing, China
{zheye,fuha,pinlin}@microsoft.com
[4] University of Montreal, Montreal, China
nie@iro.umontreal.ca

Abstract. Mining opinion is essential for consistency and persona of a chatbot. However, mining existing opinions suffers from data sparsity. Toward a given entity, we cannot always find a proper sentence that expresses desired sentiment. In this paper, we propose to generate opinion sentences for a given attitude, i.e., an entity and sentiment polarity pair. We extract attributes of a target entity from a knowledge base and specific keywords from its description. The attributes and keywords are integrated with knowledge graph embeddings, and fed into an encoder-decoder generation framework. We also propose an auxiliary task that predicts attributes using the generated sentences, aiming to avoid common opinions. Experimental results indicate that our approach significantly outperforms baselines in automatic and human evaluation.

Keywords: Opinion · Generation · Chatbot · Knowledge

1 Introduction

Conversation systems have advanced in recent years due to the progress of deep learning techniques and the accumulation of conversation data on the Internet. However, it is challenging for a conversation system to produce responses that are consistent with a specified persona. [17] found that 92.3% persona profiles and 49.2% sentences of persona profiles in PersonaChat study [20] contain at least one sentiment word[1] such as *like*, *enjoy*, and *hate*. This indicates that opinions of a given attitude are in demand in personalising chatbots and ensuring consistency.

Mining existing opinions is a way but with some issues. As Fig. 1) shows, the number of opinions of an entity is imbalanced. 1/3 entities have less than 10 opinions and 1/3 entities do not have any negative opinion. For new entities,

[1] www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon.

X. Zhu et al. (Eds.): NLPCC 2020, LNAI 12430, pp. 294–305, 2020.
https://doi.org/10.1007/978-3-030-60450-9_24

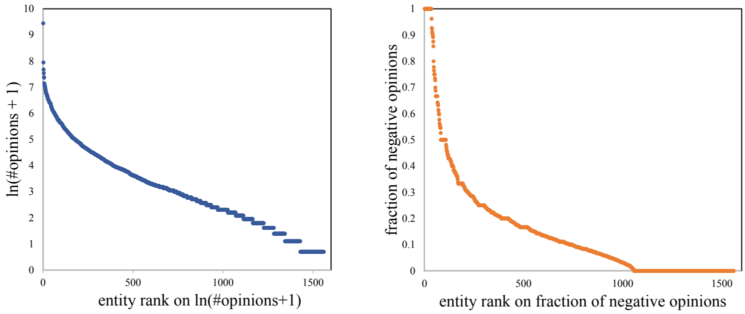

Fig. 1. Left: Number of opinions per entity. **Right:** Fraction of negative opinions per entity.

one cannot find any opinion in an existing corpus. In contrast, human can easily adopt opinions from similar entities to express their feelings about the new ones. Generation-based models provide flexibility to address the above issues. Knowledge about entities and relations between them may help.

In this paper, we propose a new way of generating opinion sentences from a given attitude. For example, from **Shaquille O'Neal** and positive sentiment polarity, we aim to generate more specific opinions like "[entity] is the forever star on the NBA All-Star stage.", rather than common opinions like "[entity] is good."

Some previous studies propose to generate opinions in specific domains. For example, the approach in [3] generates reviews of a book for a given user and rating. Within this "book" domain, generation patterns learned from one book can be easily transferred to another one. We propose a more generic method of generating opinions in mixed domains, where the entities could be persons, cities, TV series, novels, games, etc. We enhance the model's ability of transferring by incorporating knowledge base, where similar entities can be identified by their attributes. Moreover, we improve the specificity of generated opinions to avoid common but boring ones.

In this paper, we propose a new generic framework of using knowledge to generate opinion sentences from an attitude. We first represent an entity target by its general attributes in a knowledge base and specific keywords extracted from its description. Then we integrate knowledge graph embedding into the encoder-decoder framework to generate opinion sentences. Next, we propose using an auxiliary task of using opinion sentences to predict attribute values to enhance specificity. Evaluation indicates that our proposed approach significantly outperforms baselines in generating more interesting and specific opinion sentences.

2 Related Work

2.1 Opinion Mining

There is a long history of opinion mining or sentiment analysis. As [10] described, opinion mining aims to identify and extract subjective content in text and thus most works focus on sentiment classification. What we address is not classification but generation. Some works studied generation, e.g. concept-to-text generation. For example, [12] generates weather forecast or sports reports from structured input data. They regard the input data and output sentences as sequences and apply RNN based encoder-decoder framework to address the problem. These works are similar to ours in structured data input and basic framework, but we have different goals. Concept-to-text tasks require the output sentences to convey the information represented by the input data. There are relatively limited templates that can be mapped to the given schema of database. In contrast, the generated opinions have more forms. It has more serious one-to-many issue. What we need is to generate appropriate and specific opinion sentences to express a chatbot's persona. [3] proposes a new task of generating reviews from a triple of userID, productID, and rating. Their goal is close to ours but they conduct experiments on one category of products, i.e., books. They do not leverage knowledge graph to extend the entity perhaps because their data is relatively rich for the same category.

2.2 Generation Models

There are many generation models are proposed for sequence-to-sequence generation. The two main applications are machine translation and conversation generation. Our task is more relevant to conversation generation because our input also has many proper outputs (one-to-many). We share the same issue that common results are much more easily generated but they lack of information and diversity[8,16,19]. To solve the issue, we propose using $P(X|Y)$, where X is the source sequence and Y is the target sequence, to balance the frequency and the specificity. The main idea is similar to that used by [9,18] but in different ways due to different tasks. To the best of our knowledge, we are the first who introduce the similar methods into opinion generation. Similar to [9], we face similar practical issues when we try to integrate $P(X|Y)$ into objective function: intractable models and ungrammatical generation. We propose our own solution to solve these issues.

2.3 Knowledge Graphs

A typical knowledge graph contains millions of triples (h, r, t) where h is the head entity, t is the tail entity and r is the relation from h to t. Knowledge graph embedding models learn low-dimensional representation vectors for entities or relations. The embedding preserves the structural information of the original knowledge graph. Translation-based knowledge graph embedding models, such

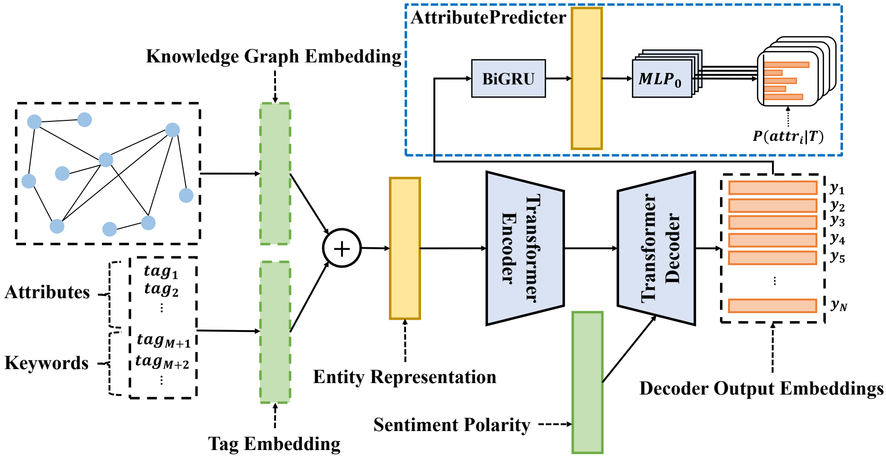

Fig. 2. The overview of **KN**owledge Enhanced **O**pinion **G**eneration Model.

as TransE, are proved effective [1]. Recently, graph neural network (GNN) has attracted a lot of attention. Graph Convolutional Networks (GCN) [7], as one of GNNs, can be used to encode the graph information. The GCN showed promising performance in graph node classification tasks [5] and semantic role labeling task [11]. We apply GCN to embed knowledge graphs in our approach.

3 Problem Formulation

Given an attitude, i.e., an entity e and its sentiment polarity $p \in \{+1, -1\}$, the task is to generate opinions of the entity e that express the sentiment polarity p. The generated opinions are expected to be 1) fluent, 2)coherent with the sentiment polarity, 3) relevant and specific to the entity e.

A training sample is a triple (e, p, Y), which denotes that the sentence $Y = [y_0, y_1, ..., y_N]$ expresses a sentiment polarity of p toward the entity e. We also make use of a knowledge graph G. The graph contains three types of nodes, namely entities, attribute values, and keywords. Nodes are connected by different types of edges, which correspond to keyword and different types of attribute. For each entity node, there is a corresponding description document d.

4 Our Approach

As Fig. 2 shows, we propose a generic framework to solve the problem. We provide details in this section. We first describe how to represent an entity as the input of our model. Then we integrate a knowledge graph with the encoder-decoder framework. At last we describe how to improve the quality of generated opinions by avoiding common opinion sentences.

4.1 Entity Representation

An entity itself does not provide much information for generation. We extend the representation of an entity by its attribute values in a knowledge base G. For example, the entity of **Shaquille O'Neal** has attributes like entity type, nationality, gender, occupation. The entity can be represented as [person, American, male, basketball player, ...]. **Song Xu** is a Chinese singer and the representation could be [person, Chinese, male, singer, ...].

We find the attributes are not specific enough for entities. For example, many basketball players have the same attribute values like [person, American, male, basketball player, ...]. This results in common generation results. We further extend the representation of an entity by keywords extracted from its description document d. We extract the top k frequent keywords (excluding stop words). For example, **Shaquille O'Neal** has keywords like *NBA*, *center*, *star*. **Tim Duncan** has keywords like *Spurs*, *history*, and *champion*. These keywords clearly distinguish the two basketball players. Therefore, an entity e is represented as

$$X = (attr_1, attr_2, ...attr_M, keyword_1, keyword_2, ..., keyword_K)$$

and we let the number of dimension of X $T = M + K$. In our experiments, $M = 11, K = 10$.

4.2 Encoder-Decoder Framework with Knowledge Graph Integrated

Encoder-Decoder Framework
We choose a transformer-based encoder-decoder model as a start. We define $\mathbf{e}_i(\cdot)$, $\mathbf{e}_o(\cdot)$ and $\mathbf{e}_s(\cdot)$ as three functions for looking up embeddings for inputs, outputs and sentiment polarities.

An L-layer transformer is used as the encoder. Given the input sequence $X = (attr_1, attr_2, ..., attr_M, keyword_1, keyword_2, ..., keyword_K)$, we first pack their embeddings into

$$H_0 = [\mathbf{e}_i(attr_1), ..., \mathbf{e}_i(attr_T), \mathbf{e}_i(keyword_1), ..., \mathbf{e}_i(keyword_K)]. \quad (1)$$

The output of the last layer $H = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_T]$ are used as the encoded representation vectors of the X which are calculated by

$$\mathbf{H} = Trans_e^L(\mathbf{H}_0). \quad (2)$$

where $Trans_e^L(\cdot)$ represents the transformer encoder.

Another L-layer transformer is used as the decoder. The decoding procedure of i-th step is as follows:

$$\mathbf{E}_i = [\mathbf{e}_o(y_0), \mathbf{e}_o(y_1), ..., \mathbf{e}_o(y_{i-1})] \quad (3)$$

$$\mathbf{S}_i = Trans_d^L(\mathbf{H}, \mathbf{E}_i, \mathbf{e}_s(p)) \quad (4)$$

where $Trans_d^L(\cdot)$ is the L-layer transformer decoder, $\mathbf{E}_i$ is composed of embeddings of decoded words. $\mathbf{S}_i$ is composed of i output embeddings, $[\mathbf{s}_1^i, \mathbf{s}_2^i, ..., \mathbf{s}_i^i]$, in i-th step. The pilot experiments show the sentiment signal may fade away during broadcasting from the encoder to the decoder. Then the generated opinion sentences are poorly coherent to the sentiment. We feed the sentiment polarity p to the decoder in every step instead of treating it as an input of the encoder.

The unnormalized generation probability $P(y_i)$ is conditioned on the output embedding $\mathbf{s}_i^i$:

$$\begin{aligned} P(y_i = w) &= P_V(y_i = w) \\ &= \mathbf{w}^T \cdot MLP_V(\mathbf{s}_i^i) \end{aligned} \tag{5}$$

where $\mathbf{w}$ is the one-hot indicator for word w.

Integrating Knowledge Graph Embedding

In order to further leverage the knowledge graph as a whole, we propose using knowledge graph embeddings to represent the attribute values of an entity e. We incorporate Graph Convolutional Network (GCN), which is a neural network model designed for graph-structured data [7], into our opinion generation model.

$\mathbf{e}_g(attr_i)$, $\mathbf{e}_g(keyword_j)$ is the graph embeddings of $attr_i$ and $keyword_j$. We use a linear transformation to merge the graph embedding with the original tag embedding $\mathbf{e_i}(attr_i)$ as follows:

$$\mathbf{e}_m(attr_i) = \mathbf{W}^T[\mathbf{e}_g(attr_i); \mathbf{e}_i(attr_i)] \tag{6}$$

Then we replace the $\mathbf{e}_i(attr_i)$ of Eq. 1 with the $\mathbf{e}_m(attr_i)$ to encode graph information into the opinion generation model. We update the parameters of GCN along with the parameters of the main model.

4.3 Promoting Specificity by Enhancing Knowledge

In dialogue generation, generation-based models tend to generate common responses. A common response can be coherent to many different input utterances [8,19]. An opinion generation model based on a vanilla encoder-decoder framework also suffers from generating common opinions. A common opinion sentence is coherent to many different entities. We can use the pattern "[entity] is good" to generate "[O'Neal] is good" and "[Paris] is good". On the contrary, "[entity] is the forever star on the NBA All-Star stage" is a specific opinion. One can infer that it is used to express a positive sentiment about a NBA basketball player. If the generation model knows the specific degree of an opinion sentence, it will be able to avoid from generating common opinion sentences.

Inspired by the recent studies on the diversity and specificity in dialogue generation task [8,14,16,19,21], we propose our methods to improve our opinion generation model by promoting specificity with the help of knowledge information (attributes). The main idea is to predict attribute values based on a generated opinion sentence. The attribute values shall be accurately predicted for an opinion sentence with good specificity. We use cross entropy to measure the difference between the predicted attribute distribution and the ground-truth attribute distribution. A small difference means the attribute prediction model can easily infer the ground-truth attribute values. It further indicates the given opinion sentence is specific. So the calculation procedure of specificity of Y is as follows:

$$spec(Y) = \exp(-\sum_{i=1}^{M} P(attr_i|Y)ln(\hat{P}(attr_i|Y))) = \prod_{i=1}^{M} \hat{P}(attr_i|Y) \tag{7}$$

where $P(attr_i|Y)$ is the ground-truth distribution of $attr_i$ and $\hat{P}(attr_i|Y)$ is the predicted distribution of $attr_i$. Because $attr_i$ is the true attribute value of Y's entity, $P(attr_i|Y)$ equals 1.

It is intuitive that if the model can "see" more specific training samples and less common training samples, the model will tend to generate the specific opinion sentences. We assign every training sample a sampling probability. Before every training epoch, we re-sample the training dataset to get a new training dataset with the same size according to the sampling probabilities. A training sample with higher sampling probability has more chances to be seen by the model. We use $spec(Y)$ as the sampling probability of an opinion sentence Y. We use Bi-directional GRUs to encode opinion sentence Y,

$$\mathbf{h}_t^o = BiGRU(\mathbf{e}_i(y_t), \mathbf{h}_{t-1}^o); t \in [1, N] \tag{8}$$

and then use M (the number of attributes) softmax-based classifiers to get attribute distributions, $\hat{P}(attr_1|Y)$, $\hat{P}(attr_1|Y)$, ... $\hat{P}(attr_M|Y)$. The model gives "[entity] is the forever star on the NBA All-Star stage" a score of 0.997 and "[entity] is good" a score of 0.021.

Joint Learning: We regard the opinion generation as the main task and the attribute prediction task as the auxiliary task. Applying joint learning is supposed to increase the specificity of generated opinion sentences. But if we take the generated opinion sentences as the input to the attribute prediction model, the training procedure is intractable. So we use the decoder output embeddings $\mathbf{s}_1^1, \mathbf{s}_2^2, ..., \mathbf{s}_{|N|}^{|N|}$ of opinion generation model as the representation of input opinions to attribute prediction models (See upper right part of Fig. 2):

$$\mathbf{h}_t^o = BiGRU(\mathbf{s}_t^t, \mathbf{h}_{t-1}^o); t \in [1, N]. \tag{9}$$

Then the convergence of the auxiliary task could "force" the main model to produce more specific opinion sentences. We denote the attribute prediction distribution as $\hat{P}'(attr_i|Y)$.

Re-ranking: When performing decoding, we use beam search to find all candidates according to the scores from the main model. After that, we re-rank all candidates by a combination of specific degree and the main model output scores as follows:

$$score(\hat{Y}) = log(P(\hat{Y}|X)) + \alpha \sum_{i=1}^{M} \hat{P}(attr_i|Y) + \beta \sum_{i=1}^{M} \hat{P}'(attr_i|Y). \tag{10}$$

Table 1. The statistics on the data for the experiments.

	#Entity	#Attitude	#Opinion	#Labeled
TRAIN	1,314	2141	104,823	—
DEV	100	162	7,751	—
TEST.SEEN	61	61	2,484	—
TEST.UNSEEN	130	217	7,386	—
TEST.SEEN.HUMAN	18	18	—	1310
TEST.UNSEEN.HUMAN	18	30	—	2227

5 Experiment

5.1 Dataset

To construct training samples, we use a pre-trained attitude detector [17] to detect sentiment polarity p and associated entity e from a Chinese conversation corpus. Responses with positive or negative attitude were kept as opinion sentences. To obtain the sentence Y, the entity e in an opinion sentence were replaced with a special token `[entity]`. In this way, we obtained triples like (e, p, Y) for training.

We split the data into four parts (see Table 1). Entities in TEST.UNSEEN were not included in either TRAIN or DEV. Entities in TEST.SEEN were included in TRAIN with the opposite polarity. Due to the cost, we selected 30 attitudes from TEST.UNSEEN denoted as TEST.UNSEEN.HUMAN and another 18 attitudes from TEST.SEEN denoted as TEST.SEEN.HUMAN for human evaluation. The top ten generated opinions from different methods were pooled together and shuffled before showing them to every annotator. Even though, an annotator had to labeled more than 3500 opinions.

5.2 Baselines

Retrieval: Given an entity E_i and the expected polarity, we find the entity E_j with the most similarity with E_i from the training data. We choose N opinions with descending similarity. We define the similarity between two entities as the weighted sum of the matched attributes. We give larger weight to more important attribute.

LibFM: A recommendation model [13] is used to "recommend" opinions for the given entity and sentiment polarity. Embeddings of the sentiment, attributes, keywords, graph nodes and words in opinions are used as the side information.

Att2Seq: We adopt the model proposed by [3] to generate opinions conditioned on the attitude polarity and the attributes of an entity.

Transformer: We use a 6-layer transformer as the encoder and another 6-layer transformer as the decoder. The whole structure is similar to [15].

Table 2. The second, third and fourth column show the ratios of generated opinions with overall scores of **+2**, **+1** and **0**. **Spec** column shows the ratio of specific opinions. **Avg** column is the average overall score based on the human evaluation scores. **nDCG** column is used to show the quality of the generated opinions from the view of ranking over the all models generated opinions list. The **bold** means the model outperforms all other models in term of that metric. * indicates KNOG outperforms Transformer significantly ($p < 0.05$).

Model	+2	+1	0	Spec	Avg	nDCG	NIST	Dist-1	Dist-2
Retrieval	0.172	0.475	0.353	0.309	0.818	0.376	0.714	**0.114**	**0.347**
LibFM	0.099	0.412	0.490	0.216	0.609	0.274	0.009	0.034	0.095
Att2Seq+A	0.193	0.565	**0.242**	0.324	0.951	0.443	0.505	0.040	0.174
Transformer	0.156	0.572	0.272	0.247	0.885	0.409	0.470	0.038	0.165
KNOG	**0.279**	0.458	0.263	**0.420**	**1.017***	**0.484***	**1.240**	0.047	0.205
vs.Att2Seq+A	↑ 44.6%	↓ 18.8%	↑ 8.3%	↑ 29.8%	↑ 6.9%	↑ 9.3%	↑ 145.6%	↑ 18.8%	↑ 17.9%
vs.Transformer	↑ 78.7%	↓ 19.9%	↓ 3.3%	↑ 69.9%	↑ 14.9%	↑ 18.5%	↑ 163.9%	↑ 23.9%	↑ 24.6%
-Reranking	0.239	0.463	0.298	0.367	0.941	0.441	1.051	0.049	0.208
-Joint learning	0.238	0.467	0.295	0.340	0.943	0.449	0.772	0.044	0.184
-Reranking	0.223	0.460	0.317	0.326	0.906	0.432	0.861	0.045	0.183

5.3 Experiment Settings

Our models and baselines are implemented by PyTorch[2]. The sizes of embeddings and hidden states in our encoder-decoder framework are set to 768. We use 1-layer bidirectional GRUs with a hidden size of 768 to encode the opinions for predicting the distribution of attributes. We also use another 1-layer bidirectional GRUs with a hidden size of 768 to encode the decoder output embeddings of the generated opinions in joint learning. We tune the α and β based on the performance of our model on the Dev in terms of the automatic metrics. We use Adam optimizer [6] to train models with learning rate of 1e-4. Except LibFM, all other trainable models are trained for 50 epochs. LibFM are trained for 100 epochs because it needs more epochs training to converge.

5.4 Evaluation Methodology

We conduct automatic and human evaluations to compare our approach with baselines. In automatic evaluation, we employ NIST [2], Distinct-1 and Distinct-2 [8] as metrics. NIST and BLEU are two variants of N-gram scoring metrics which are widely used in machine translation. NIST gives larger weights to those N-grams which are more informative. Distinct-1 and Distinct-2 are used to measure the diversity of generated sentences based on the ratio of unique unigrams and bigrams.

In human evaluation, we recruited three human annotators, who are independent of authors and are not major in computer science. Each sentence is judged by the following criteria:

[2] https://pytorch.org.

Table 3. Generated results by **Transformer** (left) and our knowledge enhanced model **KNOG** (right).

Song Xu:[person, China, male, singer/actor]		
1	I like [Song Xu] 我喜欢[许嵩]	Who I like is [Song Xu] 我喜欢的人是[许嵩]
2	I love [Song Xu] 我爱[许嵩]	I wanna listen the songs of [Song Xu] 我喜欢听[许嵩]的歌
3	[Song Xu] is a good singer [许嵩]是个好歌手	[Song Xu]'s voice is the best [许嵩]的声音是最棒的
Hunan:[place, China]		
1	I wanna go to [Hunan] 我想去[湖南]	I wanna go to [Hunan] 我想去[湖南]
2	I am going to [Hunan] 我打算去[湖南]	[Human]'s weather is good. Missing [湖南]的天气真好，怀念ing
3	I like [Hunan] 我喜欢[湖南]	[Hunan] is a good place [湖南]是个好地方
Final Fantastic:[game]		
1	I like [Final Fantastic] 我喜欢[最终幻想]	I like playing [Final Fantastic] 我喜欢玩[最终幻想]
2	[Final Fantastic] is interesting [最终幻想]是有趣的	I like the man who plays [Final Fantastic] 我喜欢玩[最终幻想]的人
3	[Final Fantastic] is very interesting [最终幻想]非常有趣	[Final Fantastic] is funny [最终幻想]很逗

- **Good (+2):** The sentence is fluent. The opinion exactly expresses the given attitude. And the opinion is interesting and appropriate.
- **Fair (+1):** The sentence is fluent. The opinion exactly contains the given attitude. The opinion is not interesting.
- **Bad (+0):** The sentence cannot be understood. Or the generated opinion is not consistent with the given attitude or not reasonable in terms of facts.

The annotators also judged whether a sentence is specific or not. The annotators completed the two tasks with Fleiss' kappa [4] of 0.379 and 0.411, which means fair and moderate agreement respectively.

5.5 Result and Analysis

Table 2 shows all experimental results. The Att2Seq outperforms other baseline models by generating more **good** opinions and fewer **bad** opinions. In human evaluation, our model KNOG outperforms Att2Seq by 6.9% and 9.3% improvements in terms of average score and nDCG. In automatic evaluation, our model also significantly outperforms Att2Seq in terms of NIST, Distinct-1 and Distinct-2 by 145.6%, 18.8% and 17.9%. Our model is based on Transformer. And our model also outperforms Transformer in terms of those metrics used in our experiment. KNOG can generate 78.7% more high quality opinion sentences which are labeled as **Good (+2)** and 3.3% fewer **Bad** opinion sentences. The knowledge enhancement also makes the model can generate more specific, interesting and coherent opinions. The comparison between KNOG and Transformer shows that our model can actually promoting the diversity of generated opinions.

In order to further study Reranking and Joint Learning's impact, we do an ablation study. The last three rows of Table 2 shows the ablation result. We can find that Reranking and Joint learning both can make the model generate more **Good** opinion sentences and fewer **Bad** opinion sentences. And combining them can enhance these effects. It seems reranking can improve the model in terms of NIST, Avg and nDCG but slightly deteriorate the diversity. In general, reranking can also improve the overall performance.

Table 3 shows some cases that are generated by a baseline and KNOG. We can find KNOG would generate more specific opinions. It can generate detailed attributes of the entity.

6 Conclusion and Future Work

In this paper, we propose a knowledge enhanced opinion generation model based on the transformer-based encoder-decoder model to address the problem of generating opinion sentences by a given attitude. We leverage a knowledge base and descriptions to extend entity names to tags and integrate knowledge graph embedding methods into our model to further exploit knowledge graph. Moreover, we propose to use reranking and joint learning to enhance the knowledge in generated opinions. Experimental results shows that using our model would improve the generated opinions significantly by increasing **Good** opinions and decreasing **Bad** opinions at the same time. As future work, we plan to investigate how to combine knowledge graph with the main model more closely.

References

1. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held 5–8 December 2013, Lake Tahoe, Nevada, United States, pp. 2787–2795 (2013)
2. Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the Second International Conference on Human Language Technology Research, pp. 138–145. Morgan Kaufmann Publishers Inc. (2002)
3. Dong, L., Huang, S., Wei, F., Lapata, M., Zhou, M., Xu, K.: Learning to generate product reviews from attributes. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp. 623–632 (2017)
4. Fleiss, J.L., Cohen, J.: The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. Educ. Psychol. Measur. **33**(3), 613–619 (1973)
5. Hammond, D.K., Vandergheynst, P., Gribonval, R.: Wavelets on graphs via spectral graph theory. CoRR abs/0912.3848 (2009)

6. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. CoRR abs/1412.6980 (2014)
7. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. CoRR abs/1609.02907 (2016)
8. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. abs/1510.03055 (2015). http://arxiv.org/abs/1510.03055
9. Li, J., Galley, M., Brockett, C., Spithourakis, G.P., Gao, J., Dolan, W.B.: A persona-based neural conversation model. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016, Volume 1: Long Papers (2016)
10. Liu, B.: Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press, Cambridge (2015)
11. Marcheggiani, D., Titov, I.: Encoding sentences with graph convolutional networks for semantic role labeling. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, 9–11 September 2017, pp. 1506–1515 (2017)
12. Mei, H., Bansal, M., Walter, M.R.: What to talk about and how? Selective generation using LSTMs with coarse-to-fine alignment. arXiv preprint arXiv:1509.00838 (2015)
13. Rendle, S.: Factorization machines with libFM. ACM Trans. Intell. Syst. Technol. **3**(3), 57:1–57:22 (2012)
14. Shen, X., et al.: A conditional variational framework for dialog generation. abs/1705.00316 (2017). http://arxiv.org/abs/1705.00316
15. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017, pp. 6000–6010 (2017)
16. Xing, C., et al.: Topic aware neural response generation. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA, 4–9 February 2017, pp. 3351–3357 (2017)
17. Zeng, Z., Song, R., Lin, P., Sakai, T.: Attitude detection for one-round conversation: jointly extracting target-polarity pairs. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 285–293. ACM (2019)
18. Zhang, H., Lan, Y., Guo, J., Xu, J., Cheng, X.: Reinforcing coherence for sequence to sequence model in dialogue generation. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, 13–19 July 2018, pp. 4567–4573. ijcai.org (2018)
19. Zhang, R., Guo, J., Fan, Y., Lan, Y., Xu, J., Cheng, X.: Learning to control the specificity in neural response generation. In: Gurevych, I., Miyao, Y. (eds.) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July 2018, Volume 1: Long Papers, pp. 1108–1117. Association for Computational Linguistics (2018)
20. Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., Weston, J.: Personalizing dialogue agents: I have a dog, do you have pets too? abs/1801.07243 (2018). http://arxiv.org/abs/1801.07243
21. Zhou, G., Luo, P., Cao, R., Lin, F., Chen, B., He, Q.: Mechanism-aware neural machine for dialogue response generation. In: Singh, S.P., Markovitch, S. (eds.) Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, California, USA, 4–9 February 2017, pp. 3400–3407. AAAI Press (2017)