SynDG: Syntax-aware Dialogue Generation

Junyan Qiu*
qiujunyan2018@ia.ac.cn
University of Chinese Academy of Sciences
Haidian District, Beijing, China

Haidong Zhang
Institute of Automation, Chinese Academy of Sciences
Haidian District, Beijing, China

Yiping Yang
Institute of Automation, Chinese Academy of Sciences
Haidian District, Beijing, China

ABSTRACT

Dialogue system is designed to converse with humans in a natural way. As an essential part of dialogue system, dialogue generation aims to generate proper response given historical context. Recently, sequence-to-sequence (seq2seq) based models have achieved great success but suffer from ungrammatical problems. In this paper, we propose a **Syn**tax-aware **D**ialogue **G**eneration (SynDG) model that incorporates syntactic information to generate grammatical responses with an encoder-decoder framework. Specifically, we first construct a syntax-graph with a dependency parser on the dialogue corpus. Then, we employ three graph embedding algorithms to learn syntactic word representations as the input of seq2seq framework. Furthermore, we devise training strategies to predict syntactic structure of the sentence for sufficient syntax understanding. Our empirical study on two multi-turn dialogue datasets demonstrates the effectiveness of SynDG in generating natural and grammatical responses.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**.

KEYWORDS

dialogue system, natural language generation, dependency parsing, graph attention network.

ACM Reference Format:
Junyan Qiu, Haidong Zhang, and Yiping Yang. 2023. SynDG: Syntax-aware Dialogue Generation. In *2023 9th International Conference on Computing and Artificial Intelligence (ICCAI 2023), March 17–20, 2023, Tianjin, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3594315.3594364

1 INTRODUCTION

Open domain dialogue system aims at providing entertainment for users by generating "human-like" responses. It has a wide range of applications on social chatbots such as Microsoft XiaoIce [20] and Amazon Alexa [17]. Thanks to the rapid advancements in sequence-to-sequence (seq2seq) modeling techniques, data-driven approaches have drawn increasing research attentions and achieved significant progress recently.

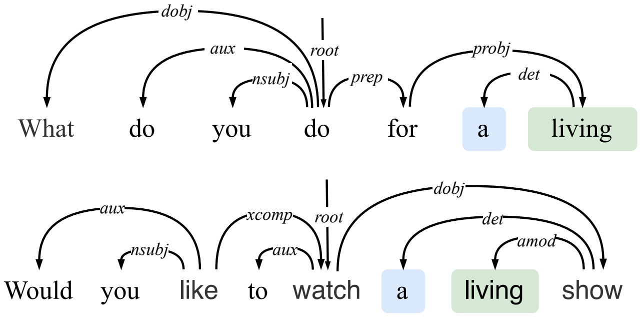


Figure 1: Examples of dependency parsing. The dependencies between tokens *"a"* and *"living"* vary given different contexts.

Despite all these achievements, seq2seq models face the challenges of being generic and ungrammatical. Most previous works focused on addressing the former issue by grounding the generation in external information [18, 31] or proposing new objective functions to measure the mutual dependence between the context and responses [8]. However, very little is studied to consider word's syntactic properties, which is pre-requisite for the system to converse with users smoothly.

In order to generate grammatical responses, it is important to disclose the way that words are combined to constitute sentences and how the information of sentences is governed [30]. Dependency parsing (DP), referring to uncovering the internal structural relations between tokens in a sequence, is normally employed to analyze the grammatical structure of sentence. For example, [13] extracted graph-structured features derived from dependency parsing to evaluate the quality of machine translation. [29] treated argument mining (AM) as a dependency parsing problem to analyze the argument structure for a sentence in a neural end-to-end manner. [28] framed factuality assessment as a model dependency parsing task to identify the events and their sources.

In this paper, we develop a **Syn**tax-aware **D**ialogue **G**eneration network (SynDG) upon the seq2seq framework for grammatical response generation. To be specific, motivated by [13], a syntactic graph is constructed on the whole dialogue corpus using a dependency parser to capture relations between words of input sentences. In the graph, vertices and edges represent tokens and dependencies between them respectively. Then, we employ three graph embedding learning algorithms, including TransE, TransR and graph attention network (GAT), to learn word representations, which are fed into the seq2seq framework to generate responses. Unfortunately, the static word embeddings can not capture the

[*] Both authors contributed equally to this research.

ICCAI 2023, March 17–20, 2023, Tianjin, China

ACM ISBN 978-1-4503-9902-9/23/03.
https://doi.org/10.1145/3594315.3594364

dynamic syntactic relationships between tokens across various contexts, as shown in Fig. 1. Thus, we introduce training strategies with two subtasks, including dependency labels and directional edges prediction, to pre-train the model together with embeddings. To conclude, our contributions can be summarized as follows.

- We propose a novel model named SynGAT that integrates dependency parsing and graph embedding learning algorithms to learn syntactic word representations for grammatical dialogue generation.
- We introduce new training strategies to model complex syntactic dependency relations between words. To the best of our knowledge, this is the first work that apply dependency parsing for dialogue generation.
- We conduct comprehensive experiments on two multi-turn dialogue datasets Holl-E and DailyDialog. Experimental results and further ablation studies demonstrate the effectiveness of our proposed method.

2 RELATED WORK

2.1 Dialogue system

Dialogue systems have attracted sensational attentions in both academy and industry communities with a wide range of application prospects, such as Eliza [27], Alice [4] and Microsoft XiaoIce [20]. Early systems applying hand-crafted rules to imitate human behaviors are constrained to certain environment [11]. Recently, there emerges a new trend that models the dialogue generation as a seq2seq framework and train it in a end-to-end manner. However, these methods still suffer from generic and non-informative problems. To address that, various approaches have been proposed including diversity enhancement [5, 8], enlarging model scales [3, 16] and grounding external knowledge [18, 31]. Nevertheless, how to generate grammatical sentences remains to be a challenge. One approach is to manually label incorrect sentences and train the model to detect and correct ungrammatical pieces, which is expensive and time-consuming. The other approach is to produce synthetic instances using deep learning techniques [21, 22]. Both approaches overlook the syntactic information. Besides, the performance is highly related to the quality of annotated dataset. In this paper, we integrate dependency parsing and graph attention networks to learn syntactic word representations for grammatical response generation without requiring extra ungrammatical annotations.

2.2 Graph embedding learning

Graph is a kind of non-Euclidean data structure with rich relation (edges) information among objects (nodes). Recently, there is a large number of applications that represent data in the form of graphs, such as biology, citations and social networks. Graph neural network (GNN), which employs deep learning algorithms to address graph-related tasks in an end-to-end manner, mainly focuses on tasks including node classification, link prediction and clustering. Recently, there emerges a lot of graph learning works that embed discrete vertices into a continuous vector space. For example, TransE [2] regarded relations as translations that connect head and tail entities in the embedding space. However, TransE can only handle one-to-one mappings while has difficulty in dealing with one-to-many mappings. Thus, [26] introduced TransH that models relations as a hyperplane, on which the translation from head to tail is operated. [10] further proposed to build entity and relations in distinct spaces to extend modeling flexibility. As a special architecture of GNN, graph attention network (GAT) [24] computed node representations in the graph by attending over its neighbors, allowing the most important part to be focused. Previous researches usually applied recurrent neural network (RNN) or transformer [23] to model sequences, which is effective when operating on regular Euclidean space while is incapable of dealing with graph data. In this paper, to incorporate syntax information into response generation, we construct a syntactic graph and employ three graph embedding learning algorithms to learn word representations.

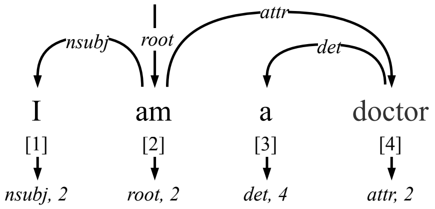


Figure 3: Dependency parsing for sentence “I am a doctor”.

3 SYNTAX-AWARE DIALOGUE GENERATION

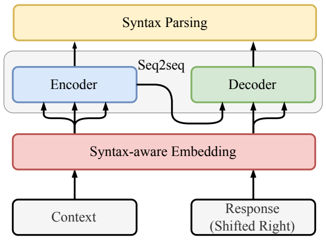


Figure 2: Overview architecture of SynDG.

The overview architecture of our proposed method is presented in Fig. 2, which can be mainly divided into three components, including (1) syntax-aware embedding converting dialogue inputs to into dense vectors, (2) seq2seq generating responses given encoded context and (3) syntax parsing predicting dependencies among words in a sentence. In the following section, we will introduce them systematically.

3.1 Syntax-aware embedding

In order to extract syntactic information between words in a sentence, we build syntax-aware embedding in the following steps.

Given a dialogue corpus, all words in the vocabulary are regarded as a vertex of the graph. The edges are represented as the syntactic relationships between words in dependency trees, as shown in Fig. 3. The graph is stored in the form of triplets (h, r, t), where h and t short for head and tail are words in the vocabulary, r short for relation is dependency relation label predefined in the parser. For

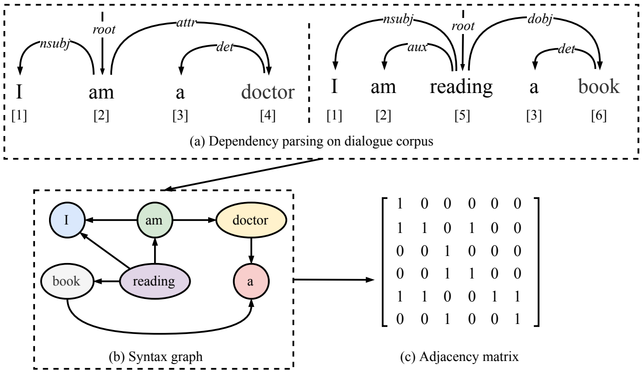


Figure 4: Adjacency matrix construction

example, as shown in Fig. 3, the sentence *"I am a doctor"* can be transformed to three triplets, i.e., $(am, nsubj, I)$, $(am, attr, doctor)$ and $(doctor, det, a)$.

In this paper, we employ three graph embedding algorithms, including TransE [2], TransR [10] and graph attention network (GAT) [24], to learn syntax-aware word representations.

TransE. TransE represents a relation by a translation vector $\mathbf{r}$ that connects two entities to form a triplet with high plausibility. Concretely, if a (h, r, t) is a golden triplet that exists in the graph $\mathcal{G}$, then the embedding of the tail entity $\mathbf{t}$ can be transformed to the embedding of the head entity $\mathbf{h}$ by adding $\mathbf{r}$, i.e., $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$, otherwise the opposite. To this end, the objective is to minimize the following margin-based ranking criterion over the training set:

$$\mathcal{L} = \sum_{(h,r,t)\in\mathcal{G}} \sum_{(h',r,t')\notin\mathcal{G}} [\gamma + f_r(\mathbf{h},\mathbf{t}) - f_r(\mathbf{h}',\mathbf{t}')]_+ \tag{1}$$

where $\gamma > 0$ is a hyper-parameter, $[x]_+ \triangleq \max(0, x)$, $f_r(\mathbf{h},\mathbf{t}) = \|\mathbf{h}+\mathbf{r}-\mathbf{t}\|_{\ell_2}$. (h', r, t') is a corrupted triplet constructed by negative sampling.

TransR. One major problem of TransE is that it models embeddings of the entities and relations in the same space, which is insufficient for modeling. To address that issue, TransR proposes to build entity and relation in distinct spaces by projecting entities from entity space to relation space and connecting the projected entities with relations. Mathematically, the score function is defined as:

$$f_r(\mathbf{h},\mathbf{t}) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_{\ell_2} \tag{2}$$

where $\mathbf{h}_r = \mathbf{h}\mathbf{M}_r$, $\mathbf{t}_r = \mathbf{t}\mathbf{M}_r$, $\mathbf{M}_r$ is a projection matrix.

GAT. As shown in Fig. 4, given all triplets of the dialogue corpus, we convert them into adjacent matrix $\mathbf{A} \in \mathbb{R}^{|V|\times|V|}$ as follows:

$$a_{ij} = \begin{cases} 1 & i = j \text{ or } (v_i, v_j) \in \mathcal{G} \\ 0 & (v_i, v_j) \notin \mathcal{G} \end{cases} \tag{3}$$

where $|V|$ is the number of vertex in the graph, a_{ij} is the value at the i_{th} row, j_{th} column in $\mathbf{A}$. v_i is the i_{th} node, (v_i, v_j) denotes the edge from v_i to v_j.

After obtaining the adjacency matrix, we discard the relation between two entities and represent each node embedding $\mathbf{h}_i \in \mathbb{R}^{d_m}$ by attending over its neighbors.

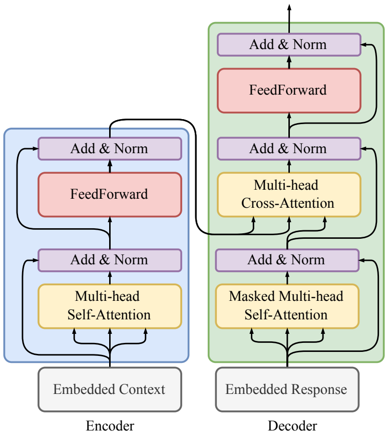


Figure 5: Architecture of transformer.

$$\mathbf{h}_i = \tanh(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\mathbf{W}\mathbf{h}_j) \tag{4}$$

$$\alpha_{ij} = \frac{\exp(\mathbf{h}_i^T \mathbf{h}_j)}{\sum_{k\in\mathcal{N}_i} \exp(\mathbf{h}_i^T \mathbf{h}_k)} \tag{5}$$

where d_m is the dimension of embeddings, $\mathcal{N}_i$ denotes the neighbor of node i.

3.2 Seq2seq

The seq2seq framework is composed of a encoder and a decoder, which are implemented by transformer [23]. As shown in Fig. 5, the encoder is composed of two sub-layers. The first sub-layer is the multi-head self-attention [23] that captures semantic dependencies and the second one is a feedforward neural network (FFN). In addition to the aforementioned two sub-layers, the decoder inserts a multi-head cross attention sub-layer between them, which calculates attention distributions from response to the output of encoder. For the first sub-layer, the decoder masks future tokens to prevent information from flowing to subsequent positions. Notably, each sub-layer is followed by a residual connection [6] and layer normalization [1].

Given the dialogue context $\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$ with n words (utterances from different turns will be concatenated by a special toke [SEP]), we first map them into dense vectors $\mathbf{x}_i \in \mathbb{R}^{d_m}$ using the syntax-aware embeddings. Then the encoder takes these vectors as input and generates semantic hidden representations $\{\mathbf{h}_1^x, \mathbf{h}_2^x, \cdots, \mathbf{h}_n^x\}$, which are fed into the decoder to calculate the response representations:

$$\mathbf{h}_i^r = \text{Transformer}(\mathbf{h}_{<i}^r, \mathbf{h}_{1\sim n}^x) \tag{6}$$

$\mathbf{h}_i^r$ is then used to compute the probability distribution over the vocabulary $\mathcal{V}$ through a softmax function. We leverage cross entropy between the generated response and true label y_i as the training objective.

$$p(\tilde{y}_i|y_{<i},\mathcal{X}) = \mathrm{softmax}(\mathbf{W}^{vocab}\mathbf{h}_i^r + \mathbf{b}^{vocab}) \tag{7}$$

$$\mathcal{L}_{GEN} = \sum_i -y_i \log(p(\tilde{y}_i|y_{<i},\mathcal{X})) \tag{8}$$

3.3 Syntax parsing

Although syntax-aware embedding contains the syntactic information, it can not model the various syntactic dependencies between words across different linguistic contexts. To address that issue, we propose novel training strategies that decompose syntax parsing into two subtasks including dependency label and directional edge prediction, and train the embedding together with the seq2seq framework. The two subtasks are designed to predict the relation and head for each word, which plays the role of the tail a triplet. In this way, the syntactic information is encoded as a function of the whole input sentence and can capture context-dependent word relations.

Dependency labels prediction. Considering the syntax graph of a sentence, the indegree of a vertex is no more than one, i.e., a word can be the tail of at most one triplet. Based on this observation, we define the dependency label of a word w as the relation r, where w is the tail of a triplet connected by r. As show in Fig. 3, the dependency label for word *"I"* is *"nsubj"* since it is the tail of triplet $(am, nsubj, I)$. Assuming there are N labels predefined in the parser.

To better retrieve syntactic information from the sentence, we develop a classifier to predict dependency labels. Specifically, given the hidden word representations of the context $\mathbf{h}_i^x$ (or response $\mathbf{h}_i^r$) , we exploit a feedforward neural network with ReLU activation function to calculate probability distributions over the dependency label set.

$$[a_1, a_2, \cdots, a_N] = \mathbf{W}_2[\mathrm{ReLU}(\mathbf{W}_1\mathbf{h}_i^x + \mathbf{b}_1)] + \mathbf{b}_2 \tag{9}$$

$$p_i = \frac{\exp(a_i)}{\sum_{j=1}^{N}\exp(a_j)} \tag{10}$$

where $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$ are trainable parameters. We adopt cross entropy loss to quantify the difference between the true label y_i^{dep} and the generated probabilities.

$$\mathcal{L}_{DEP} = \sum_{i=1}^{N} -y_i^{dep}\log(p_i) \tag{11}$$

Directional edges prediction. Directional edges indicate path from the head to tail. Now that we obtain the relation, our goal is to assign a head for each tail. In this paper, we employ pointer network [25] that selects word corresponding to position as the head from the input sentence. As shown in Fig. 3, the head of word *"I"* is *"am"* at position 2. Given the hidden representation $\mathbf{h}_i$ ($\mathbf{h}_i^x$ or $\mathbf{h}_i^r$), we apply an attention mechanism to calculate probability distribution over the input sentence.

$$q_{ij} = \frac{\exp(\mathbf{h}_i^T\mathbf{h}_j^x)}{\sum_{k=1}^{\ell}\exp(\mathbf{h}_i^T\mathbf{h}_k)} \tag{12}$$

where ℓ is the length of the input sentence, q_{ij} indicates the probability of word j being the head of word i. The training objective is defined as the cross entropy between the probabilities and true position labels $y_{ij}^{pos} = \begin{cases} 1 & \text{if word } j \text{ is the head of word } i \\ 0 & \text{otherwise} \end{cases}$.

$$\mathcal{L}_{POS} = \sum_{j=1}^{\ell}\sum_{i=1}^{\ell} -y_{ij}^{pos}\log(q_{ij}) \tag{13}$$

The final loss is defined as:

$$\mathcal{L} = \mathcal{L}_{GEN} + \mathcal{L}_{DEP} + \mathcal{L}_{POS} \tag{14}$$

4 EXPERIMENTS

4.1 Datasets and experiment details

Table 1: Statistics of DailyDialog and Holl-E datasets.

Datasets	Holl-E	DailyDialog
#dialogues	9069	13118
#utterances	91438	102980
avg. words per utterance	15.14	14.45
train/dev/test	34486/4318/4388	11118/1000/1000
#triplet	411191	295562
#vertices/words	27051	17034

We employ two datasets, namely Holl-E [12] and DailyDialog [9] to demonstrate the effectiveness of our proposed method. Holl-E is a chit-chat dataset between two interlocutors chatting about movies given plots, reviews or commons. The are more than 9k dialogues an 90k utterances. In this paper, the background knowledge is ignored. DailyDialog is a multi-turn dialogue dataset for exchanging information and enhancing social bonding. It contains more than 13k dialogues with rich intention and emotion information. The detail statistics of two datasets are presented in Table 1.

In this paper, the seq2seq framework is constructed by transformer with 6 layers, 300 hidden nodes and 6 attention heads. The maximum length of the context and response is set to 256 and 64 respectively. Text that exceeds the length limits will be truncated. During training, adam [7] is leveraged to optimized model parameters with learning rate annealing from 10^{-4} to 10^{-5}. Following [13], we leverage the NLP library spaCy to parse sentences, which contains 45 dependency labels predefined in the parser. There are approximately 411k and 295k triplets in the syntax graph constructed from Holl-E and DailyDialog respectively. The dimension of syntax-aware embedding is set to 300 compatible with the subsequent seq2seq framework.

Perplexity (**PPL**) and **BLEU** [14] are used to evaluate the coherent and fluency of generated responses. We also adopt human judgemets to alleviate the limits of automatic evaluation metrics. Concretely, we randomly sample 50 cases and rate them from 1 to 5 in terms of grammatical accuracy (**GRAM**), where responses with higher score indicate higher confidence of being grammatically correct.

Table 2: Experimental results of SynDG and other baselines on datasets Holl-E and DailyDialog.

Models	Holl-E			DailyDialog		
	PPL	BLEU	GRAM	PPL	BLEU	GRAM
Seq2seq	73.14	7.25	2.37	54.51	37.19	3.17
Seq2seq+copy	67.35	8.45	2.89	51.76	39.81	3.31
Seq2seq+GloVe	63.19	8.19	3.19	48.19	38.94	3.61
HRED	75.86	7.68	3.27	55.67	39.77	3.51
SynDG(TranE)	61.45	8.98	**3.51**	47.17	43.11	3.81
SynDG(TransR)	**59.18**	**10.56**	3.45	**44.52**	**44.29**	**3.92**
SynDG(GAT)	62.15	8.78	3.19	49.81	42.18	3.84

4.2 Comparison models

To demonstrate the effectiveness of our method, we conduct contrast experiments on these models:

- **Seq2seq:** It is the vanilla sequence-to-sequence framework applying attention mechanisms.
- **Seq2seq+copy:** Seq2seq framework with copy mechanism [25].
- **Seq2seq+GloVe:** The embedding of seq2seq framework is initialized by GloVe [15].
- **HRED:** A hierarchical encoder-decoder model that encodes dialogue context in both word level and utterance level [19].

4.3 Experiment results

The main experimental results are shown in Table 2 and we have the following observations: (1) It is very clear that our proposed SynDG outperforms other baselines in both automatic and manual evaluations. The overall performance supports the declaration that our method facilitates generating high-quality responses. (2) Particularly, compared to seq2seq+GloVe, which also applies pre-trained word embedding for efficient semantic understanding, our method, e.g., SynDG(TransR), obtains impressive 3.67, 4.65% and 0.31 improvements on DailyDialog in terms of PPL, BLEU and GRAM respectively. This is because our syntax-aware method benefits a lot from syntactic information by graph learning. (3) Last block of Table 2 presents the results of three graph learning algorithms applied to train word embeddings. Generally, TransR achieves the best results. It is explainable that it models entities and relations in distinct spaces, which is more sufficient for modeling than TransE. Moreover, GAT lags behind the other two since it ignores explicit relations between words.

4.4 Ablation study

To further demonstrate the effectiveness of each component, we conduct several ablation experiments on DailyDialog and present the results in Table 3. It shows that both syntax-aware embedding and syntax parsing can improve the performance. An interesting phenomenon can be observed from the results. For automatic metrics (i.e., PPL and BLEU), removing syntax-aware embedding results in more severe performance drop. While larger degradation can be observed after discarding syntax parsing in terms of GRAM. More specifically, after we remove dependency labels or directional edges prediction, it reduces the GRAM by 0.31 and 0.29, which causes competitive performance decline with casting off the whole syntax parsing (0.35). It demonstrates that both of them are crucial in generating grammatical responses.

Table 3: The ablation study of syntax-aware embedding and syntax parsing on DailyDialog dataset. wo. means without, pred. means prediction.

Models	PPL	BLUE	GRAM
SynDG(TransR)	44.52	44.29	3.92
wo. syntax-aware embedding	45.13	42.19	3.77
wo. syntax parsing	47.19	42.77	3.51
wo. dependency labels pred.	46.89	42.52	3.63
wo. directional edges pred.	46.31	43.51	3.57

4.5 Case study

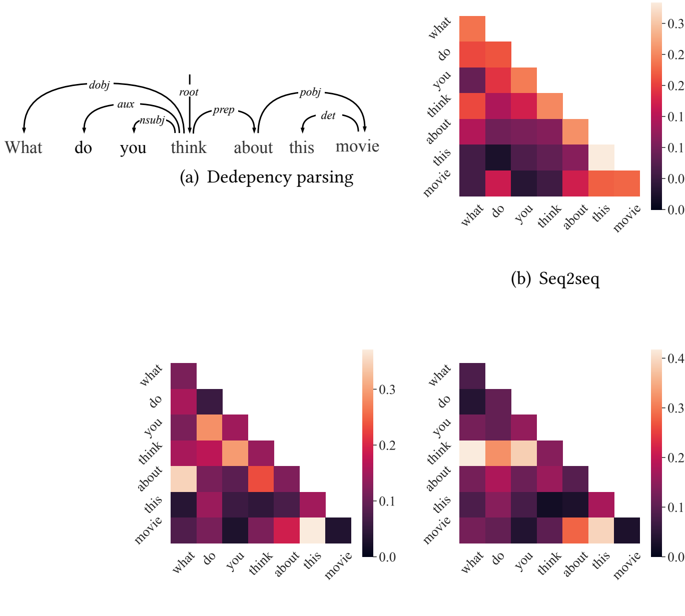
Table 4: Dialogue examples of golden responses and those generated by seq2seq and SynGD models on test dataset of DailyDialog.

Context	u_1 : I'm sorry I'm so late, ... u_2 : It's ten after six. But dinner is at six thirty.
Golden:	I know, I'm really sorry. I lost my bag.
Seq2seq	Sorry, sorry, sorry.
SynGD	I feel so sorry.
Context	u_1 : There is a new gial in school, have you seen her yet?
Golden	I haven't seen her yet.
Seq2seq	All right, are you?
SynGD	No, I haven't, what about you?

We present two dialogue examples generated by different models in Table 4. It is clear that SynGD can generate grammatical and coherent responses. Comparatively, seq2seq model tend to generate responses with ungrammatical and reduplicated pieces (case 1) or responses that are inconsistent with the context (case 2).

4.6 Dependency visualization

To study how our method facilitates modeling syntactic information, we present Fig. 6, which illustrates the visualization of word dependencies from seq2seq and different components of SynDG. From Fig. 6, we can observe that (1) Seq2seq exhibits week word relations that accords with the syntactic structure of the sentence. (2) Syntax-aware embedding establishes strong correlations between phrases with high occurrence, e.g., *"what"* and *"about"*, *"this"* and *"movie"* and so on. (3) SynDG can accurately capture the word dependencies according to the syntactic structure of the sentence. For example, (*"what"*, *"think"*), (*"do"*,*"think"*), (*"about"*,*"movie"*) and so on.



(a) Dedepency parsing

(b) Seq2seq

(c) Syntax-aware embedding

(d) SynDG (TransR)

Figure 6: Visualizations of word dependencies for sentence *"What do you think about this movie"*. The visualizations are obtained by calculating the cosine similarity between (b) outputs of the seq2seq encoder, (c) syntax-aware embedding and (d) outputs of SynDG, of corresponding words.

5 CONCLUSION

In this paper, we introduce a novel method named SynDG that incorporates syntactic information for grammatical dialogue generation. Concretely, we first construct a syntax graph by a dependency parser and employ three graph leaning algorithms to learn word embeddings. Then we develop two training strategies upon the sequence-to-sequence framework, which is trained together with the syntax-aware embedding, to learn word dependencies between words. Extensive experiments show that our proposed method can generate more grammatical as well as coherent responses. We also conduct ablations experiments to demonstrate the effectiveness of each component.

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[4] Alice E Fischer and Frances S Grodzinsky. 1993. *The anatomy of programming languages.* Prentice Hall.

[5] Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and William B Dolan. 2019. Jointly Optimizing Diversity and Relevance in Neural Response Generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 1229–1238.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[7] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[8] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 110–119.

[9] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 986–995.

[10] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

[11] Longxuan Ma, Mingda Li, Wei-Nan Zhang, Jiapeng Li, and Ting Liu. 2021. Unstructured Text Enhanced Open-Domain Dialogue System: A Systematic Survey. *ACM Transactions on Information Systems (TOIS)* 40, 1 (2021), 1–44.

[12] Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M Khapra. 2018. Towards Exploiting Background Knowledge for Building Conversation Systems. In *EMNLP*.

[13] Bin Ni, Xiaolei Lu, and Yiqi Tong. 2021. SynXLM-R: Syntax-Enhanced XLM-R in Translation Quality Estimation. In *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer, 27–40.

[14] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[15] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[16] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[17] Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604* (2018).

[18] Pengjie Ren, Zhumin Chen, Christof Monz, Jun Ma, and Maarten de Rijke. 2020. Thinking globally, acting locally: Distantly supervised global-to-local knowledge selection for background based conversation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 8697–8704.

[19] Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[20] Heung-Yeung Shum, Xiao-dong He, and Di Li. 2018. From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering* 19, 1 (2018), 10–26.

[21] Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. *arXiv preprint arXiv:2105.13318* (2021).

[22] Yujin Takahashi, Satoru Katsumata, and Mamoru Komachi. 2020. Grammatical error correction using pseudo learner corpus considering learner's error tendency. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. 27–32.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. *Advances in Neural Information Processing Systems* 30 (2017), 5998–6008.

[24] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[25] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Advances in neural information processing systems* 28 (2015).

[26] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 28.

[27] Joseph Weizenbaum. 1966. ELIZA—a computer program for the study of natural language communication between man and machine. *Commun. ACM* 9, 1 (1966), 36–45.

[28] Jiarui Yao, Haoling Qiu, Jin Zhao, Bonan Min, and Nianwen Xue. 2021. Factuality assessment as modal dependency parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 1540–1550.

[29] Yuxiao Ye and Simone Teufel. 2021. End-to-end argument mining as biaffine dependency parsing. In *Proceedings of the 16th Conference of the European Chapter*

of the Association for Computational Linguistics: Main Volume. 669–678.

[30] Meishan Zhang. 2020. A survey of syntactic-semantic parsing based on constituent and dependency structures. *Science China Technological Sciences* 63, 10 (2020), 1898–1920.

[31] Wen Zheng and Ke Zhou. 2019. Enhancing conversational dialogue models with grounded knowledge. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 709–718.