2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)

Precise Description Generation for Knowledge Base Entities with Local Pointer Network

Shyh-Horng Yeh,[1] Hen-Hsen Huang,[1] and Hsin-Hsi Chen[12]
[1]Department of Computer Science and Information Engineering, Taipei, Taiwan
[2]MOST Joint Research Center for AI Technology and All Vista Health care, Taipei, Taiwan
{shyeh, hhhuang}@nlg.csie.ntu.edu.tw, hhchen@ntu.edu.tw

***Abstract*—Verbalization of knowledge base (KB) facts about an entity allows users to absorb information from KB more easily. The drawback of most previous work is that they cannot generalize to unseen frames. This work introduces the task of precise KB verbalization that is aimed at generating an exact description for the given factual triples. We propose a novel sequence-to-sequence (seq2seq) model with the local pointer network to deal with this task. The approach to training data construction is also explored. Experimental results show our method improves the performances in terms of Meteor and slot error rates. Human evaluation is also performed to confirm the effectiveness of our model.**

Keywords—knowledge base, natural language generation, pointer network.

I. Introduction

Knowledge bases (KBs) like Freebase, DBpedia, and Wikidata have become the backbone for a variety of applications such as question answering [1]. Facts stored in most KBs are usually in the triple form (*subject*, *predicate*, *object*) where subject and object are two entities in the KB, and predicate is a relation between the two entities. In other words, a KB is a graph, and entities and relations are vertexes and edges in the graph, respectively. The structural organization of facts in KBs make them easily accessible by the machine. A variety of graph based and social network approaches has been proposed for different applications. On the other hand, the facts directly retrieved from the KB are unfriendly to human. For example, (S1) shows three facts returned from a KB by querying information about an entity "Bruce Springsteen". Compared with the factual triples, a description in natural language like (S2) provides the information in a much more intuitive way for end users. Accordingly, an information system that is able to verbalize the knowledge in the KB allows users to absorb information friendlier and more quickly. This work is aimed at verbalization of KB entities. Given a target entity *e* and a set of factual triples, where *e* is the subject of each triple, our goal is to generate a one-sentence description such that it conveys the exact meaning of the factual triples.

(S1) (Bruce Springsteen, people.person.profession, singer-songwriter) (Bruce Springsteen, music.artist.genre, rock) (Bruce Springsteen, music.artist.album, Born to Run)

(S2) Bruce Springsteen is a pop singer-songwriter known for his album "Born to Run."

Instead of the template based or the retrieval based approach, we form our goal as a generation task. That is, the description can be freely generated by our model given arbitrary factual triples. For this reason, we aim to train a natural language generative model that learns to map the knowledge in KB to natural language descriptions. We propose a sequence-to-sequence (seq2seq) neural network [2] for the conversion from a set of structural data to an unstructured description in English.

Neural network-based seq2seq models have shown their effectiveness in natural language generation like machine translation and dialogue system. These models rely on large amount of training data, and the availability of quality training data for this task is a main challenging issue in our work. Specifically, the dataset that can be used for model training should consist of large amounts of pairs between the natural language descriptions and the factual triples to entail the descriptions. Due to the lack of ready-made datasets, we propose a method based on the distance supervision strategy to construct the training pairs without manual annotation.

Another challenging issue in this work is the requirement to exactly express the given factual triples. Additional facts that are not expressed in the given factual triples are undesired. In other words, our model is capable of generating precise descriptions in accordance with the given fact triples. Thus, the outcome of our model is easily controllable by specifying the desired triple set. To achieve this goal, we propose a local pointer network model, which is shown to be effective to our task.

The contributions of this work are threefold: (1) This work is the first attempt to address the task of precise KB verbalization. The challenging issues of this task are identified and discussed. (2) We propose a novel and effective neural network model for this task. The results are confirmed with human verification. (3) Our strategy for training data construction can be applied to the related tasks.

The rest of this paper is organized as follows. Section II summaries the related work of the verbalization of KBs. We also briefly review the progress of seq2seq models in natural language processing. In Section III, we introduce the task of precise KB verbalization and present a generative model for this task. Section IV describes the method for dataset construction, which is an important step for training our model. Experimental results are discussed in Section V. Finally, Section VI concludes this work.

II. Related Work

On natural language generation for KB content, some of the previous work require manually-constructed lexicon [3], [4]. Other work that does not involve human labor can be roughly categorized into template-based approach or retrieval-based approach.

The template-based approach works by extracting frequent patterns of each frame. The sentences are generated by filling in the patterns. The work of Li et al. [5] identify the aspects of a sentence and extract frequent sub-dependency

978-1-5386-7325-6/18/$31.00 ©2018 IEEE
DOI 10.1109/WI.2018.00-87

trees of each aspect. These frequent tree patterns are then converted back to sentence patterns. Unlike the work of Li et al. [5], which extract tree patterns, Duma and Klein [6] extract word-sequence patterns. Ell and Harth [7] represent a sentence's content as an RDF graph pattern. Frequent maximal subgraph pattern mining is then used to extract pairs of (sentence, subgraph patterns). Voskarides et al. [8] take into consideration the knowledge graph of the target entity and select the template that best fits with the entity semantically using a learning-to-rank approach. The main problem of the template-based approach is that they cannot generalize to unseen frames.

The works based on the retrieval-based approach retrieve sentences from the documents of the target entities. Sauper and Barzilay [9] extract templates and use the sentences generated by template filling to query a search engine for relevant excerpts. A selection model then selects excerpts that can form a coherent summary of the target entity. A bootstrapped approach is proposed by Saldanha et al. [10] to generate description for companies. Voskarides et al. [11] retrieve candidate sentences containing two entities from a search engine and choose the one that best describes the relationship between the two entities. The drawback of the retrieval-based approach is the requirement of a set of documents of the target entity. Thus it may not be suitable for domain-specific entities or emergent entities.

Some of the proposed methods are able to compose new templates for unseen frame, mitigating the problems of both template-based and retrieval-based methods. Gyawali and Gardent [12] propose a method based on Feature-Based Lexicalized Tree Adjoining Grammar (FB-LTAG) with a unification-based semantics. It combines the sub-trees of each slot with FB-LTAG operations, so it is able to generate descriptions for unseen frames. The system achieved a state-of-the-art performance in the KBgen challenge dataset [13]. Lebret et al. [14] and Chisholm et al. [15] jointly learn a content selection model and surface realizer to generate one sentence for person entities from Wikipedia infobox and Wikidata.

Unlike the work of Gyawali and Gardent [12] we construct our corpus without manual annotation. The methods we proposed in this work can generalize to unseen frames. Different from the work by Lebret et al. [14] and Chisholm et al. [15], our task requires the generated sentences covey exactly the meaning of the frame, while the prior work is to express an arbitrary subset of the frame as long as the slots in the subset are coherent.

Recently, seq2seq neural network models are explored to KB verbalization. In the preliminary studies, the vanilla seq2seq models are adopted [16], [17]. A seq2seq network translates a source sequence to a target sequence, which naturally fits a variety of tasks in natural language processing and achieves effectiveness results in many of these tasks. The common choices for the recurrent unit inside the encoder and the decoder of seq2seq network are Long Short-Term Memory [18] and Gated Recurrent Unit [19]. The attention mechanism in seq2seq networks plays an important role in some tasks like machine translation and summarization [20]. The attention mechanism calculates an attention score for each hidden state of the encoder, and sums the hidden states with the attention scores as weights. Seq2seq models with attention mechanism are able to attend to the important part of the input sequence and thus improve its ability to fulfill the task.

Vinyals et al. [21] and Gu et al. [22] proposed variants of seq2seq models based on the idea of attention mechanism. Pointer networks use the attention scores as weights to copy elements from the sequence [21]. The output dictionary of a pointer network is of variable size that depends on the input. Gu et al. [22] incorporates this idea into machine summarization models with the copy mechanism. The copy mechanism either generates a word from a vocabulary or copies a word from the input sequence. Thus, it is able to generate translations with rare words.

III. Generative Model for Precise KB Verbalization

In this work, the generation task is dealt with by a seq2seq model. The goal of our task is to generate a description given a set of factual triples. However, it will be extremely sparse directly learning to convert the triples to description. Instead of end-to-end learning, our generative model is trained to convert a frame, which is a set of predicates, to a descriptive template. Specifically, the input of our models is the frame derived from the given factual triples, and the output is a template with a number of slots. The slots will then be filled with the values specified by the factual triples. This process is illustrated in Fig. 1. To generate a precise description, the model is expected to have the following characteristics:

- It should *not* generate slots in the template that are not specified in the given frame.
- Each slot in the frame can only be generated *exactly once*.
- The generated description must *exactly* describe the slots given in the frame.

Section III.A describes a vanilla seq2seq model, which is taken as our baseline. Since vanilla seq2seq models are not guaranteed to hold these characteristics, in Sections III.B and III.C, we introduce our dedicated seq2seq architecture elaborated for this task with pointer networks.

A. Vanilla Sequence to Sequence Neural Networks

The basic seq2seq neural network models $P(Y \mid X)$ with an encoder and a decoder. The encoder is a recurrent neural network (RNN) that encodes the source sequence $X = (x_1, x_2, \dots x_N)$ into a vector $Enc(X)$, which is assumed to be a representation that encodes the syntactic and semantic information of X to a vector space. The decoder is another RNN that generates a target sequence $Y = (y_1, y_2, \dots y_M)$ conditioned on $Enc(X)$. There are several choices for the recurrent unit of RNN, among which LSTM and GRU are common choices and have been shown effective. In this work, the recurrent unit of the encoder and the decoder is LSTM.

Formally, given a sequence $X = (x_1, x_2, \dots x_N)$, an encoder of a RNN based on LSTM cells is denoted as $Enc(X)$. $P(Y \mid X)$ can be decomposed into factors with product rule:

$$P(Y \mid X) = P(y_1, y_2, \dots y_M \mid X) = \prod_{t}^{M} P(y_t \mid y_1 \dots y_{t-1}, X)$$

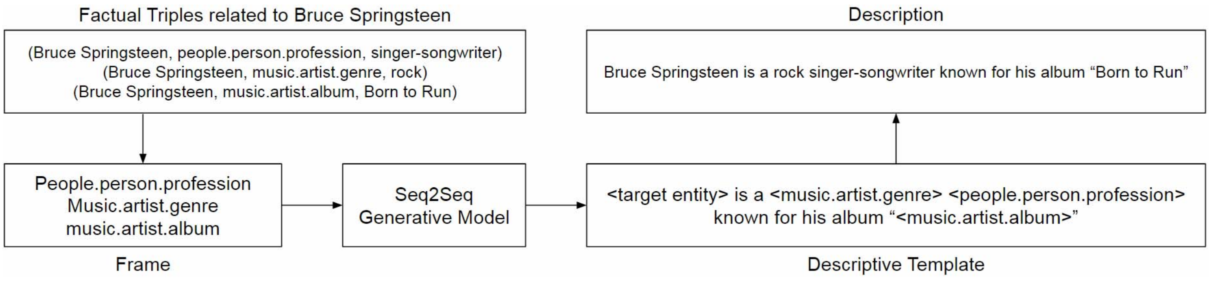

Factual Triples related to Bruce Springsteen
(Bruce Springsteen, people.person.profession, singer-songwriter)
(Bruce Springsteen, music.artist.genre, rock)
(Bruce Springsteen, music.artist.album, Born to Run)
Description
Bruce Springsteen is a rock singer-songwriter known for his album "Born to Run"
People.person.profession
Music.artist.genre
music.artist.album
Frame
Seq2Seq
Generative Model
<target entity> is a <music.artist.genre> <people.person.profession> known for his album "<music.artist.album>"
Descriptive Template

Fig. 1. KB Verbalization as a sequence-to-sequence task.

The conditioned part $y_1 \dots y_{t-1}, X$ in each factor $P(y_t \mid y_1 \dots y_{t-1}, X)$ can be again modeled by a RNN, followed by a $softmax$ function:

$$H_0 = Enc(X)$$
$$H_t = LSTM(H_{t-1}, y_t)$$
$$P(y_t \mid y_1 \dots y_{t-1}, X) = softmax(UH_t)$$

, where $LSTM(h, y)$ denotes a time-step in RNN and U is a matrix of size $V \times d_H$.

During the generation phase, the output sequence Y can be either generated by sampling one token at each time step from $P(y_t \mid y_1 \dots y_{t-1}, X)$, or by beam-searching over the output space.

Under the vanilla seq2seq framework, our input sequence X is a set of predicates to be conveyed, and the output sequence Y is a template of description. We sort the predicates according to their frequencies in the corpus in descending order to form the input sequence X. The output vocabulary V is the union of the general vocabulary $V_{general}$ and the slot vocabulary V_{slot} composed of all the possible slot terms such as <people.person.profession> and <music.artist.genre>.

B. *Sequence to Sequence Neural Networks with Global Pointer Network*

The template generated by the vanilla seq2seq model may lack some slots that are specified in the given frame. On the other hand, the template may also contain some slots that are undesired. To enforce the model to generate the templates with exact slots, our seq2seq model employs the copy mechanism so that the model is guaranteed to generate the slots only appearing in the frame.

The copy mechanism in seq2seq models is first proposed to mitigate the out-of-vocabulary (OOV) problem in neural machine translation and summarization. With such mechanism, the model will either generate words from generation vocabulary V_g (the generation mode) or copy words from the input sequence vocabulary V_c (the copy mode). Note that $V_g \cap V_c$ might not be empty. Formally, the copy mechanism assumes the equation as follows.

$$\begin{aligned} P(y_t \mid y_1 \dots y_{t-1}, X) &= P_g(y_t | y_1 \dots y_{t-1}, X) \\ &+ P_c(y_t | y_1 \dots y_{t-1}, X) \end{aligned}$$

The probability for generation mode is similar to the vanilla seq2seq:

$$P_g(y_t | y_1, y_2, \dots y_{t-1}, X) \propto \begin{cases} e^{\varphi_g(y_t)}, & if\ y_t \in V_g, \\ e^{\varphi_g(<UNK>)}, & if\ y_t \notin V_g \cup V_c \\ 0, & if\ y_t \in \overline{V_g} \cap V_c \end{cases}$$

, where $\varphi_g(w)$ is the generation score for word w. The probability for copy mode is a pointer network, in which the output prediction directly depends on the input sequence:

$$P_c(y_t | y_1, y_2, \dots y_{t-1}, X) \propto \begin{cases} \sum_{x_j = y_t} e^{\varphi_c(x_j)}, & if\ y_t \in V_c \\ 0, & otherwise \end{cases}$$

, where $\varphi_c(w)$ is the copy score for word w.

The input sequence vocabulary V_c of the seq2seq model with copy mechanism depends on the input sequence, which is suitable for our tasks because this property also holds for our slot vocabulary V_{slot}. Under the copy mechanism framework, the general vocabulary $V_{general}$ corresponds to the generation vocabulary V_g, while the input slot vocabulary V_{slot} corresponds to the input sequence vocabulary V_c. Different from the machine translation task, there are two properties of our task:

The elements in our input sequence, i.e. (predicate, slot) pairs, are unique. In other words, $V_{general} \cap V_{slot}$ is always an empty set. Consequently, the copy mechanism used in our models can be written in Equation (1) as follows.

$$P(y_t | y_1, y_2, \dots y_{t-1}, X) \propto \begin{cases} e^{\varphi_{general}(y_t)}, & if\ y_t \in V_{general} \\ e^{\varphi_{slot}(y_t)}, & if\ y_t \in V_{slot} \\ 0, & otherwise \end{cases}$$

Equation (1)

The score functions are defined as follows.

$$\varphi_{general}(y_t) = UH_t\ for\ y_t \in V_{general}$$
$$\varphi_{slot}(y_t = w) = h_{[w]} W H_t\ for\ w \in V_{slot}$$

, where $[w]$ is the position of w in the input sequence.

Our model can be viewed as a seq2seq with a *global* pointer network. The pointer network is *global* in the sense that all the source hidden states h_t are involved in calculation at each time step in the decoder. We denote this model as Seq2Seq-GPN.

C. Sequence to Sequence Neural Networks with Local Pointer Network

Although Seq2Seq-GPN always generates slots given in the frame, it still tends to output duplicate slots and ignore other slots. We further improve the Seq2Seq-GPN model by replacing the global pointer network with a *local* one. In this way, each slot is guaranteed to be generated exactly once. Specifically, we modified Equation (1) to Equation (2) as follows.

$$P(y_t|y_1, y_2, \dots y_{t-1}, X) \propto \begin{cases} e^{\varphi_{general}(y_t)}, & if\ y_t \in V_{general} \\ e^{\varphi_{slot}(y_t)}, & if\ y_t \in V_{slot} \text{ and } y_i \neq y_t \\ 0, & otherwise \end{cases}$$

Equation (2)

, where i ranges from 1 to t -1.

This modification guarantees an appeared slot word will not be generated again. Here the pointer network is local because at each time step, only the necessary subset of source hidden states h_t are used to compute the probabilities of input slots that have not appeared. Fig. 2 shows how the local pointer network works. This is model is denoted as Seq2Seq-LPN.

Table I summarizes the characteristics of the three models. While none of the model is guaranteed to generate all specified slots, experimental results provided in Section V show that our Seq2Seq-LPN model can dramatically decrease the number of missing slot errors.

TABLE I. COMPARISON OF THE VANILLA SEQ2SEQ, THE SEQ2SEQ-GPN, AND THE SEQ2SEQ-LPN MODELS.

	No undesired slots	**No duplicate slots**	**No missing slots**
Seq2Seq	Not guaranteed	Not guaranteed	Not guaranteed
Seq2Seq-GPN	Guaranteed	Not guaranteed	Not guaranteed
Seq2Seq-LPN	Guaranteed	Guaranteed	Not guaranteed

IV. TRAINING DATA CONSTRUCTION

The training instances for our work are a set C of (*frame*, *description*) pairs such that the description exactly expresses all the slots in a frame of a target entity. For constructing such a dataset, we have to identify every relation mention in the description denoting a relation between the target entity and another entity. Performing such identification task by human annotators is impractical due to the large amounts of frames and descriptions. Therefore, an automatic approach to training data construction is investigated. Section IV.A shows the linguistic resources, and Section IV.B shows the distance supervision strategy used for automatic alignment between descriptions and factual triples.

A. Linguistic Resources

We adopt Freebase as the KB and Wikipedia as the source text since the sentences in Wikipedia usually contain rich information and are less noisy than other corpora like ClueWeb. Moreover, we only extract Wikipedia pages whose topic is about people because the information about people is more complete in both Freebase and Wikipedia, comparing to other domains. The alignment between Wikipedia pages and their corresponding Freebase entities can be done by matching the Freebase relation *wikipedia.en_id* values with the id of the Wikipedia articles.

An entity can be mentioned with different surface forms. For example, Barack Obama can be mentioned with the surface forms President Obama, Obama, and Barack Hussein Obama. We collect surface forms of each entity *e* by extracting the values from Freebase relation *common.notable_for.display_name, type.object.name* and *common.topic.alias*, but a preliminary inspection shows that many of the surface forms are not covered. Hence, we further gather from Wikipedia the titles of the pages (including redirection pages), and anchor text as surface forms of *e* if the anchor links to a Wikipedia page corresponding to *e*.

B. Distant Supervision Strategy for Training Data Construction

One of solutions to training data construction is performing a relation extraction system on a corpus such as Wikipedia and ClueWeb. However, current relation extraction systems are far from perfect; the F1 score of a state-of-the-art relation extraction system *CoType* [23] on the Wiki-KBP benchmark is only 0.369. Another solution is applying the distant supervision strategy on source text. Distant supervision assumes for each factual triple *f* = (*subject*, *predicate*, *object*) in the KB, the sentences that mention both the subject and the object may describe the relation between them. With this assumption, we can obtain a large training set.

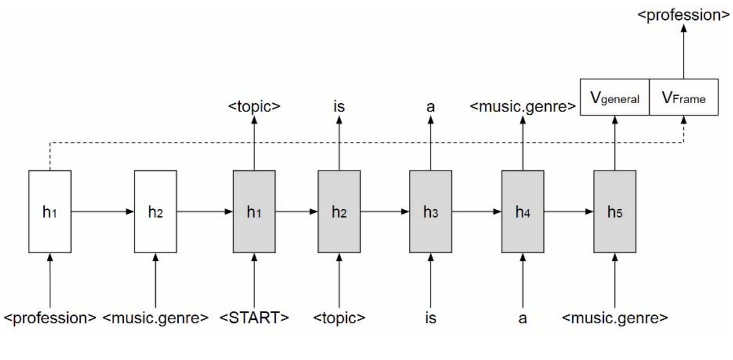
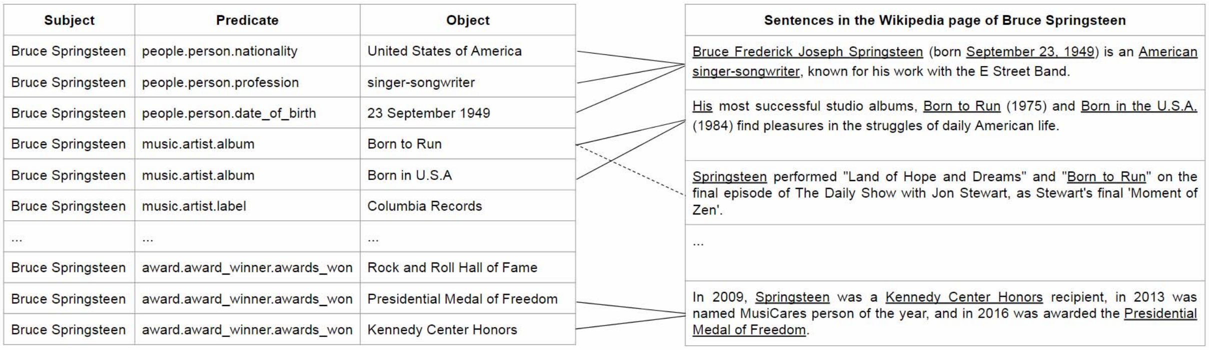


Fig. 2. The seq2seq model with the local pointer network (Seq2Seq-LPN) given the input frame in Fig. 1. Only the necessary source hidden state h_1 is involved when generating <profession>, as shown by the dotted line.



Subject	Predicate	Object
Bruce Springsteen	people.person.nationality	United States of America
Bruce Springsteen	people.person.profession	singer-songwriter
Bruce Springsteen	people.person.date_of_birth	23 September 1949
Bruce Springsteen	music.artist.album	Born to Run
Bruce Springsteen	music.artist.album	Born in U.S.A
Bruce Springsteen	music.artist.label	Columbia Records
...	...	...
Bruce Springsteen	award.award_winner.awards_won	Rock and Roll Hall of Fame
Bruce Springsteen	award.award_winner.awards_won	Presidential Medal of Freedom
Bruce Springsteen	award.award_winner.awards_won	Kennedy Center Honors

Sentences in the Wikipedia page of Bruce Springsteen
Bruce Frederick Joseph Springsteen (born September 23, 1949) is an American singer-songwriter, known for his work with the E Street Band.
His most successful studio albums, Born to Run (1975) and Born in the U.S.A. (1984) find pleasures in the struggles of daily American life.
Springsteen performed "Land of Hope and Dreams" and "Born to Run" on the final episode of The Daily Show with Jon Stewart, as Stewart's final 'Moment of Zen'.
...
In 2009, Springsteen was a Kennedy Center Honors recipient, in 2013 was named MusiCares person of the year, and in 2016 was awarded the Presidential Medal of Freedom.

Fig. 3. An example of distant supervision. The dotted line shows a wrong alignment.

Our construction method starts with distant supervision on Wikipedia. For each entity e, the factual triples (s, p, o) with $s = e$ are retrieved from Freebase. We then scan over the Wikipedia page of the entity e and extract sentences that contain both the surface form of e and the surface form of o. The process is shown in Fig. 3. Note that multiple objects may share the same surface form, so a surface form in a sentence can be linked to different o. Each relation mention in the extracted sentences is then associated with a set of candidate predicates. Here, we make a close-world assumption: only facts existing in Freebase are true while the others are regarded as false. The candidate predicates are generated by looking up the predicates between the subject and the object in Freebase.

The distant supervision strategy may introduce noisy. The dotted line in Fig. 3 shows a wrong alignment, where "Born to Run" refers to a song, rather than to an album. In addition, this sentence does not express the relation between the musician and his work because the musician can perform other's work. We reduce the impact of mis-alignment by choosing the majority predicate as the true predicate. If there are more than one majority predicate, we pick the most common one in the corpus. Finally, we further remove the sentences with duplicate slots or with the slots that do not belong to the top 100 common slots. The statistics and the frame size distribution of the training set are shown in Table II.

V. Experimental Results

Experiments are performed to evaluate the effectiveness of our model. Section V.A describes the experimental settings, and Section V.B presents the performances measured by automatic evaluation. We also invite human annotators to review the outcome of our generative model and discuss the results in Section V.C. Error analysis is given in Section V.D.

TABLE II. The statistics of the training set constructed by distant supervision.

		Frame size	Number Frames
Number of slot types	100		
Number of distinct frames	536	1	100
Number of sentences	1,335,619	2	355
Number of relation mentions	1,480,257	3	79
Number of tokens	21,420,685	4	2

A. Experimetnal Setup

For evaluation, we hold out 85 distinct frames for validation and another 85 for testing, so these hold-out frames are not seen by the model during training. We manually annotate the test data ensuring that each of the held-out frames has at most 10 correct instances. Meteor and slot error rates (extra slot rate and missing slot rate) are used as metrics. Meteor is a widely-used metric in language generation tasks such as summarization, machine translation, and dialog generation. The extra slot rate and the missing slot rate are defined as follows, respectively.

$$Extra\ slot\ rate = \frac{\#descriptions\ with\ slots\ out\ of\ the\ frame}{\#descriptions\ generated}$$

$$Missing\ slot\ error\ rate = \frac{\#descriptions\ with\ missing\ slots}{\#descriptions\ generated}$$

For each frame, the model generates 40 descriptions with beam search and the top-10 descriptions are compared to the annotated instances for evaluation.

The hyper-parameters of our model are set as follows. The sizes of the hidden states h_t and H_t are set to 256. The encoder and decoder are both 3-layer stacked LSTM. The dimension of the word embeddings of the encoder and the decoder are 100 and 150, respectively. The learning rate is set to 0.001, and the batch size is 256. We apply dropout at a rate of 0.5 between each recurrent unit of the encoder and decoder. We use negative log likelihood $L(\theta)$ as our loss function and optimize it with ADAM.

$$L(\theta) = \sum_{(X,Y)} -\log P(Y \mid X)$$

B. Results

Overall performances of the Seq2Seq, the Seq2Seq-GPN, and the Seq2Seq-LPN models are in Table III. In terms of Meteor and slot error rates, Seq2Seq-GPN outperforms the vanilla seq2seq model, and Seq2eq-LPN outperforms the other two models.

Seq2Seq-GPN never generates slots outside of the frame, but it is still possible to generate duplicate slots, causing an extra slot error. The extra slot rate of Seq2Seq-GPN is close to that of our baseline Seq2Seq. By contrast, the extra slot rate of Seq2Seq-LPN is zero in all the cases, as it is modeled in a way that prevents generating extra slots. Interestingly, although Seq2Seq-LPN is not directly modeled to generate all the slots, it has the fewest missing slot errors.

To gain more insights on how each model performs, we list the percentage of errors in frames with different number of slots in Table IV. The error rate is higher as the number of slots increases for Seq2Seq-GPN and Seq2Seq-LPN, which is reasonable because frames with more slots are more difficult to verbalize in a single sentence. Both of these models, especially Seq2Seq-LPN, are better than baseline Seq2Seq in the same number of slots, except that Seq2Seq-GPN is slightly worse than Seq2Seq when N=3. This suggests that our approach improves the ability of the seq2seq model to verbalize frames with more slots.

Even though the relation combinations of the frames in the test set do not appear in the training set, our models are still able to generate descriptions for these frames. This means our models can generate new description template out of the templates provided in the training corpus. For instance, we find that it surprisingly generates the description template for the frame {people.person.profession, music.artist.genre, music.artist.label} by combining the templates of the smaller frames {people.person.profession, music.artist.label} and {music.artist.genre, music.artist.label}. Table V shows other examples. The second example indicates the model not only combine but also prune the templates in the training corpus for unseen frames. It also learns new templates from frames with similar semantics, as shown in the third example.

TABLE III. PERFORMANCES OF THE PRECISE KB VERBALIZATION.

Model	Meteor	Extra Slots Rate	Missing Slots Rate
Seq2Seq	27.04%	23.29%	40.59%
Seq2Seq-GPN	27.58%	17.41%	23.88%
Seq2Seq-LPN	29.04%	0.00%	7.65%

TABLE IV. THE ERROR RATE WITH RESPECT TO SLOTS WITH DIFFERENT NUMBER OF SLOTS.

Model	N=1	N=2	N=3
Seq2Seq	33.57%	44.36%	33.75%
Seq2Seq-GPN	1.43%	25.64%	37.50%
Seq2Seq-LPN	1.43%	5.27%	21.25%

TABLE V. EXAMPLES ILLUSTRATING HOW THE MODELS GENERATE TEMPLATES FOR UNSEEN FRAMES.

{people.person.profession, music.artist.label} +
{music.artist.genre, music.artist.label}
→ {people.person.profession, music.artist.genre, music.artist.label}

{people.deceased_person.place_of_death,
people.deceased_person.cause_of_death}
→{people.deceased_person.cause_of_death}

{people.person.nationality, martial_arts.martial_artist.martial_art}
→{people.person.place_of_birth,
artial_arts.martial_artist.martial_art}

C. Human Evaluation

It has been pointed out in previous work that Meteor scores might not properly reflect human preferences. Hence, we conduct a human evaluation to assess the generated description of baseline Seq2Seq and our Seq2Seq-LPN models to check if the result is consistent to that of automatic evaluation.

For each frame in the held-out test set, we asked three human annotators to give scores to the two sets of top-10 descriptions generated by the two models, according to their quality of grammaticality and semantic correctness on template level and instance level. Template-level scores reflect the quality of the templates each model produces, while instance-level scores reflect the quality of sentences derived by filling the templates with actual slot values. Furthermore, since some of the generated templates contain extra information so that they cannot be used to describe a variety of knowledge base entities, we include a generality score on template level to assess whether the content of the template is too specific. All the scores are in the scale of 1 to 5. Finally, the annotators are asked to give their preference to the two models. The Fleiss's Kappa of the annotation is 0.3731, indicating a fair agreement.

The results of human evaluation on the template level are listed in Table VI. Seq2Seq-LPN has slightly higher grammaticality scores than Seq2Seq, but the p-values of 0.7055 in two-tailed Student's t-test suggest this result is insignificant. In terms of semantic correctness, Seq2Seq-LPN significantly outperforms the Seq2Seq baseline with a p-value lower than 0.005 in significant test, which is consistent with the result of automatic evaluation in Section V.B. The results on the instance level are consistent with those on the template level.

In terms of user preference, the descriptions generated by Seq2Seq-LPN are preferred over Seq2Seq in 71% of the 85 annotated frames. Both results of automatic evaluation and human evaluation indicate that our seq2seq model with the incorporation of a local pointer network can indeed make an improvement.

TABLE VI. RESULTS OF HUMAN ANNOTATION. THE SYMBOL * INDICATES THE SIGNIFICANCE.

Model	Grammaticality	Semantic Correctness	Generality
Seq2Seq	4.4039	3.1529	3.1215
Seq2Seq-LPN	4.4313	*3.6706	*3.5333

D. Error Analysis

Grammatical Errors: One of the limitations of our models is that they do not consider the slot values during training and generating, so a grammatical error might occur when the actual slot values are filled into the template. Table VII shows an instance where our model generates a grammatically correct template, but the outcome description is incorrect when the special slot value Netherlands is filled. In this case, the term "Netherlands" should be replaced with "Dutch". Re-ranking the outcomes after template-filling is a feasible solution to this kind of issues.

TABLE VII. AN EXAMPLE OF THE GRAMMATICAL ERRORS OUR MODEL PRODUCES.

Factual Triples	(Harrie van Heumen, people.person.nationaliy, Netherlands) (Harrie van Heumen, ice_hockey.hockey_player.hockey_position, ice hockey forward)
Template	<target entity> is a <people.person.nationality> <ice_hockey.hockey_player.hockey_position>
Description	Harrie is a ~~Netherlands~~ ice hockey forward.

Error Propagation from Corpus Construction: Some of the relation mentions are often wrongly labeled in the training data construction phase, and that leads the model to learn the wrong expressions for some slots. We observe that some mentions of the predicate *people.person.children* in the training set are wrongly labeled as the predicate *people.person.parent*, since the surface form of children and parents are often the same and our distant supervision method does not take text features into consideration. As a result, the model learns the wrong expression “<entity> was the father of <people.person.parents>” for the frame {*people.person.parents}* and generates a semantically incorrect template “<entity> was the father of <people.person.parents> of the <royalty.monarch.royal_line>” for the frame {*people.person.parents, royalty.monarch.royal_line*}.

Templates with Extra Information: Because of the incompleteness of Freebase, some of the information in the sentence cannot be identified. For example, in the sentence “Hiram Burgos is a <*people.person.nationality*:Puerto Rican> <*baseball.baseball_player.position_s*: baseball pitcher> who is currently a free agent.”, the relative clause “*who is currently a free agent”* is actually stating an extra fact not presented in the frame. However, this relative clause is so frequent in the training set so that the model regards it as a way to express the slot *baseball.baseball_player.position_s*. Thus, as long as an input frame contains *baseball.baseball_player.position_s*, the model will be very likely to output templates with an extra relative clause “*who is currently a free agent”.*

VI. Conclusions

This work addresses the task of precise KB verbalization. We define three criteria for generating precise description given a set of factual triples about a KB entity. By using the seq2seq approach, we propose a novel model Seq2Seq-LPN to meet the criteria. The empirical evaluation shows that the incorporation of the local pointer network is able to improve the performance in terms of Meteor and slot error rates. Although we do not force Seq2Seq-LPN to exactly generate all the slots, it still achieves a low missing slot error rate. Instead of the template base and retrieval base approaches, our generative model is capable of generating proper descriptions for unseen frames, making our model more flexible in real-world applications.

To ensure the consistency between automatic evaluation and human preferences, a human evaluation for the outcomes of baseline and our models is conducted. Seq2Seq-LPN generates better descriptions, but a semantic correctness score of 3.6 out of 5 indicates the challenge of this task. The improvement can be made by constructing less noisy training data or incorporating the adversarial learning that is expectedly more robust to noise.

Acknowledgements

This research was partially supported by Ministry of Science and Technology, Taiwan, under grants MOST-106-2923-E-002-012-MY3, MOST-107-2634-F-002-011-, and MOST-107-2218-E-009-050-.

References

[1] D. Vrandečić and M. Krötzsch. “Wikidata: A free collaborative ¨ knowledgebase.” *Commun. ACM*, vol. 57, no. 10, pp. 78–85, Sep. 2014. [Online]. Available: http://doi.acm.org/10.1145/2629489

[2] I. Sutskever, O. Vinyals, and Q. V. Le. “Sequence to sequence learning with neural networks,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112. [Online]. Available: http://papers.nips.cc/paper/5346- sequence-to-sequence-learning-with-neural-networks.pdf

[3] K. Bontcheva and Y. Wilks. “Automatic report generation from ontologies: The miakt approach,” in *Natural Language Processing and Information Systems*, F. Meziane and E. Metais, Eds. Berlin, Heidelberg: ´ Springer Berlin Heidelberg, 2004, pp. 324–335.

[4] I. Androutsopoulos, G. Lampouras, and D. Galanis. “Generating natural language descriptions from owl ontologies: the naturalowl system.” *Journal of Artificial Intelligence Research*, vol. 48, pp. 671–715, 2013.

[5] P. Li, J. Jiang, and Y. Wang. “Generating templates of entity summaries with an entity-aspect model and pattern mining,” in *Proc. 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 640–649. [Online]. Available: http://www.aclweb.org/anthology/P10-1066

[6] D. Duma and E. Klein. “Generating natural language from linked data: Unsupervised template extraction,” in *Proc. 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, 2013, pp. 83–94. [Online]. Available: http://www.aclweb.org/anthology/W13-0108

[7] B. Ell and A. Harth. “A language-independent method for the extraction of rdf verbalization templates,” in *Proc. 8th International Natural Language Generation Conference (INLG)*, 2014, pp. 26–34. [Online]. Available: http://www.aclweb.org/anthology/W14-4405

[8] N. Voskarides, E. Meij, and M. de Rijke. “Generating descriptions of entity relationships,” in *Advances in Information Retrieval*, J. M. Jose, C. Hauff, I. S. Altıngovde, D. Song, D. Albakour, S. Watt, and J. Tait, Eds. Cham: Springer International Publishing, 2017, pp. 317–330.

[9] C. Sauper and R. Barzilay. “Automatically generating wikipedia articles: A structure-aware approach,” in *Proc. Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 2009, pp. 208–216. [Online]. Available: http://www.aclweb.org/anthology/P/P09/P09-1024

[10] G. Saldanha, O. Biran, K. McKeown, and A. Gliozzo. “An entity-focused approach to generating company descriptions,” in *Proc. 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2016, pp. 243–248. [Online]. Available: http://www.aclweb.org/anthology/P16-2040

[11] N. Voskarides, E. Meij, M. Tsagkias, M. de Rijke, and W. Weerkamp. “Learning to explain entity relationships in knowledge graphs,” in *Proc. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 564– 574. [Online]. Available: http://www.aclweb.org/anthology/P15-1055

[12] B. Gyawali and C. Gardent. “Surface realisation from knowledgebases,” in *Proc. 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 424–434. [Online]. Available: http://www.aclweb.org/anthology/P14- 1040

[13] E. Banik, C. Gardent, and E. Kow. “The kbgen challenge,” in *Proc. 14th European Workshop on Natural Language Generation*, 2013, pp. 94–97. [Online]. Available: http://www.aclweb.org/anthology/W13-2111

[14] R. Lebret, D. Grangier, and M. Auli. “Neural text generation from structured data with application to the biography domain,” in *Proc. 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1203–1213. [Online]. Available: http://www.aclweb.org/anthology/D16-1128

[15] A. Chisholm, W. Radford, and B. Hachey. “Learning to generate one-sentence biographies from wikidata,” in *Proc. 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 2017, pp. 633–642. [Online]. Available: http://www.aclweb.org/anthology/E17-1060

[16] S. Agarwal and M. Dymetman. “A surprisingly effective out-of-the-box char2char model on the e2e nlg challenge dataset,” in *Proc. 18th Annual SIGdial Meeting on Discourse and Dialogue*, 2017, pp. 158–163. [Online]. Available: http://aclweb.org/anthology/W17-5519

[17] P. Vougiouklis, H. ElSahar, L. Kaffee, C. Gravier, F. Laforest, J. S. Hare, and E. Simperl. “Neural wikipedian: Generating textual summaries from knowledge base triples.” *CoRR*, vol. abs/1711.00155, 2017. [Online]. Available: http://arxiv.org/abs/1711.00155

[18] S. Hochreiter and J. Schmidhuber. “Long short-term memory.” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[19] J. Chung, C¸ . Gulc¸ehre, K. Cho, and Y. Bengio. “Empirical evaluation ¨ of gated recurrent neural networks on sequence modeling.” *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[20] D. Bahdanau, K. Cho, and Y. Bengio. "Neural machine translation by jointly learning to align and translate." *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[21] O. Vinyals, M. Fortunato, and N. Jaitly. "Pointer networks," in *Proc. 28th International Conference on Neural Information Processing Systems - Volume 2*, 2015, pp. 2692–2700. [Online]. Available: http://dl.acm.org/citation.cfm?id=2969442.2969540

[22] J. Gu, Z. Lu, H. Li, and V. O. Li. "Incorporating copying mechanism in sequence-to-sequence learning," in *Proc. 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1631–1640. [Online]. Available: http://www.aclweb.org/anthology/P16-1154

[23] X. Ren, Z. Wu, W. He, M. Qu, C. R. Voss, H. Ji, T. F. Abdelzaher, and J. Han. "Cotype: Joint extraction of typed entities and relations with knowledge bases," in *Proc. 26th International Conference on World Wide Web,* 2017, pp. 1015–1024. [Online]. Available: https://doi.org/10.1145/3038912.3052708