Zero-Shot Question Generation from Knowledge Graphs for Unseen Predicates and Entity Types

Hady Elsahar, Christophe Gravier, Frederique Laforest
Université de Lyon
Laboratoire Hubert Curien
Saint-Étienne, France
{firstname.lastname}@univ-st-etienne.fr

Abstract

We present a neural model for question generation from knowledge base triples in a "Zero-Shot" setup, that is generating questions for triples containing predicates, subject types or object types that were not seen at training time. Our model leverages triples occurrences in the natural language corpus in an encoder-decoder architecture, paired with an original part-of-speech copy action mechanism to generate questions. Benchmark and human evaluation show that our model sets a new state-of-the-art for zero-shot QG.

1 Introduction

Questions Generation (QG) from Knowledge Graphs is the task consisting in generating natural language questions given an input knowledge base (KB) triple (Serban et al., 2016). QG from knowledge graphs has shown to improve the performance of existing factoid question answering (QA) systems either by dual training or by augmenting existing training datasets (Dong et al., 2017; Khapra et al., 2017). Those methods rely on large-scale annotated datasets such as SimpleQuestions (Bordes et al., 2015). Building such datasets is a tedious task in practice, especially to obtain an unbiased dataset – i.e. a dataset that covers equally a large amount of triples in the KB. In practice many of the predicates and entity types in KB are not covered by those annotated datasets. For example 75.6% of Freebase predicates are not covered by the SimpleQuestions dataset [1]. Among those we can find important missing predicates such as: `fb:food/beer/country`, `fb:location/country/national_anthem`, `fb:astronomy/star_system/stars`.

One challenge for QG from knowledge graphs is to adapt to predicates and entity types that were *not* seen at training time (Zero-Shot Question Generation). Since state-of-the-art systems in factoid QA rely on the tremendous efforts made to create SimpleQuestions, these systems can only process questions on the subset of 24.4% of freebase predicates defined in SimpleQuestions. Previous works for factoid QG (Serban et al., 2016) claims to solve the issue of small size QA datasets. However encountering an unseen predicate / entity type will generate questions made out of random text generation for those out-of-vocabulary predicates a QG system had never seen. We go beyond this state-of-the-art by providing an original and non-trivial solution for creating a much broader set of questions for unseen predicates and entity types. Ultimately, generating questions to predicates and entity types unseen at training time will allow QA systems to cover predicates and entity types that would not have been used for QA otherwise.

Intuitively, a human who is given the task to write a question on a fact offered by a KB, would read natural language sentences where the entity or the predicate of the fact occur, and build up questions that are aligned with what he reads from both a lexical and grammatical standpoint. In this paper, we propose a model for Zero-Shot Question Generation that follows this intuitive process. In addition to the input KB triple, we feed our model with a set of textual contexts paired with the input KB triple through distant supervision. Our model derives an encoder-decoder architecture, in which the encoder encodes the input KB triple, along with a set of textual contexts into hidden representations. Those hidden representations are fed to a decoder equipped with an attention mechanism to generate an output question.

In the Zero-Shot setup, the emergence of new predicates and new class types during test time requires new lexicalizations to express these pred-

[1] replicate the observation `http://bit.ly/2GvVHae`

Proceedings of NAACL-HLT 2018, pages 218–228
New Orleans, Louisiana, June 1 - 6, 2018.

icates and classes in the output question. These lexicalizations might not be encountered by the model during training time and hence do not exist in the model vocabulary, or have been seen only few times not enough to learn a good representation for them by the model. Recent works on Text Generation tackle the rare words/unknown words problem using copy actions (Luong et al., 2015; Gülçehre et al., 2016): words with a specific position are copied from the source text to the output text – although this process is blind to the role and nature of the word in the source text. Inspired by research in open information extraction (Fader et al., 2011) and structure-content neural language models (Kiros et al., 2014), in which part-of-speech tags represent a distinctive feature when representing relations in text, we extend these positional copy actions. Instead of copying a word in a specific position in the source text, our model copies a word with a specific part-of-speech tag from the input text – we refer to those as part-of-speech copy actions. Experiments show that our model using contexts through distant supervision significantly outperforms the strongest baseline among six (+2.04 BLEU-4 score). Adding our copy action mechanism further increases this improvement (+2.39). Additionally, a human evaluation complements the comprehension of our model for edge cases; it supports the claim that the improvement brought by our copy action mechanism is even more significant than what the BLEU score suggests.

2 Related Work

QG became an essential component in many applications such as education (Heilman and Smith, 2010), tutoring (Graesser et al., 2004; Evens and Michael, 2006) and dialogue systems (Shang et al., 2015). In our paper we focus on the problem of QG from structured KB and how we can generalize it to unseen predicates and entity types. (Seyler et al., 2015) generate quiz questions from KB triples. Verbalization of entities and predicates relies on their existing labels in the KB and a dictionary. (Serban et al., 2016) use an encoder-decoder architecture with attention mechanism trained on the SimpleQuestions dataset (Bordes et al., 2015). (Dong et al., 2017) generate paraphrases of given questions to increases the performance of QA systems; paraphrases are generated relying on paraphrase datasets, neural machine translation and rule mining. (Khapra et al., 2017) generate a set of QA pairs given a KB entity. They model the problem of QG as a sequence to sequence problem by converting all the KB entities to a set of keywords. None of the previous work in QG from KB address the question of generalizing to unseen predicates and entity types.
Textual information has been used before in the Zero-Shot learning. (Socher et al., 2013) use information in pretrained word vectors for Zero-Shot visual object recognition. (Levy et al., 2017) incorporates a natural language question to the relation query to tackle Zero-Shot relation extraction problem.

Previous work in machine translation dealt with rare or unseen word problem problem for translating names and numbers in text. (Luong et al., 2015) propose a model that generates positional placeholders pointing to some words in source sentence and copy it to target sentence (*copy actions*). (Gülçehre et al., 2016; Gu et al., 2016) introduce separate trainable modules for copy actions to adapt to highly variable input sequences, for text summarization. For text generation from tables, (Lebret et al., 2016) extend positional copy actions to copy values from fields in the given table. For QG, (Serban et al., 2016) use a placeholder for the subject entity in the question to generalize to unseen entities. Their work is limited to unseen entities and does not study how they can generalize to unseen predicates and entity types.

3 Model

Let $F = \{s, p, o\}$ be the input fact provided to our model consisting of a subject s, a predicate p and an object o, and C be the set of textual contexts associated to this fact. Our goal is to learn a model that generates a sequence of T tokens $Y = y_1, y_2, \ldots, y_T$ representing a question about the subject s, where the object o is the correct answer. Our model approximates the conditional probability of the output question given an input fact $p(Y|F)$, to be the probability of the output question, given an input fact and the additional textual context C, modelled as follows:

$$p(Y|F) = \prod_{t=1}^{T} p(y_t|y_{<t}, F, C) \quad (1)$$

where $y_{<t}$ represents all previously generated tokens until time step t. Additional textual contexts are natural language representation of the triples

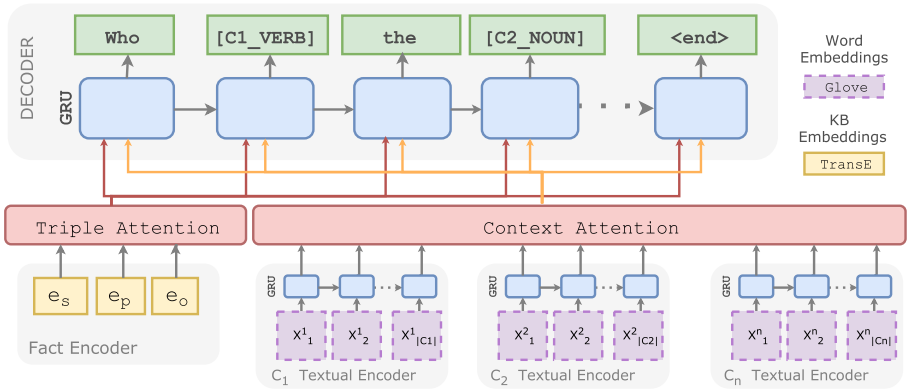

Figure 1: The proposed model for Question Generation. The model consists of a single fact encoder and n textual context encoders, each consists of a separate GRU. At each time step t, two attention vectors generated from the two attention modules are fed to the decoder to generate the next word in the output question.

that can be drawn from a corpus – our model is generic to any textual contexts that can be additionally provided, though we describe in Section 4.1 how to create such texts from Wikipedia.

Our model derives the encoder-decoder architecture of (Sutskever et al., 2014; Bahdanau et al., 2014) with two encoding modules: a feed forward architecture encodes the input triple (sec. 3.1) and a set of recurrent neural network (RNN) to encode each textual context (sec. 3.2). Our model has two attention modules (Bahdanau et al., 2014): one acts over the input triple and another acts over the input textual contexts (sec. 3.4). The decoder (sec. 3.3) is another RNN that generates the output question. At each time step, the decoder chooses to output either a word from the vocabulary or a special token indicating a copy action (sec. 3.5) from any of the textual contexts.

3.1 Fact Encoder

Given an input fact $F = \{s, p, o\}$, let each of e_s, e_p and e_o be a 1-hot vectors of size K. The fact encoder encodes each 1-hot vector into a fixed size vector $h_s = \mathbf{E_f}\, e_s$, $h_p = \mathbf{E_f}\, e_p$ and $h_o = \mathbf{E_f}\, e_o$, where $\mathbf{E_f} \in \mathbb{R}^{H_k \times K}$ is the KB embedding matrix, H_k is the size of the KB embedding and K is the size of the KB vocabulary. The *encoded fact* $h_f \in \mathbb{R}^{3H_k}$ represents the concatenation of those three vectors and we use it to initialize the decoder.

$$h_f = [h_s;\ h_p;\ h_o] \tag{2}$$

Following (Serban et al., 2016), we learn $\mathbf{E_f}$ using *TransE* (Bordes et al., 2015). We fix its weights and do not allow their update during training time.

3.2 Textual Context Encoder

Given a set of n textual contexts $C = \{c_1, c_2, \ldots, c_n\ : c_j = (x_1^j, x_2^j, \ldots, x_{|c_j|}^j)\}$, where x_i^j represents the 1-hot vector of the i^{th} token in the j^{th} textual context c_j, and $|c_j|$ is the length of the j^{th} context. We use a set of n Gated Recurrent Neural Networks (GRU) (Cho et al., 2014) to encode each of the textual concepts separately:

$$h_i^{c_j} = GRU_j\left(\mathbf{E_c}\, x_i^j,\ \ h_{i-1}^{c_j}\right) \tag{3}$$

where $h_i^{c_j} \in \mathbb{R}^{H_c}$ is the hidden state of the GRU that is equivalent to x_i^j and of size H_c . $\mathbf{E_c}$ is the input word embedding matrix. The *encoded context* represents the encoding of all the textual contexts; it is calculated as the concatenation of all the final states of all the encoded contexts:

$$h_c = [h_{|c_1|}^{c_1}; h_{|c_2|}^{c_2}; \ldots; h_{|c_n|}^{c_n}]. \tag{4}$$

3.3 Decoder

For the decoder we use another GRU with an attention mechanism (Bahdanau et al., 2014), in which the decoder hidden state $s_t \in \mathbb{R}^{H_d}$ at each time step t is calculated as:

$$s_t = z_t \circ s_{t-1} + (1 - z_t) \circ \tilde{s}_t\ , \tag{5}$$

Where:

$$\tilde{s}_t = tanh\left(W E_w y_{t-1} + U[r_t \circ s_{t-1}] + A\,[a_t^f; a_t^c]\right) \tag{6}$$

$$z_t = \sigma\left(W_z\, E_w\, y_{t-1} + U_z\, s_{t-1} + A_z\,[a_t^f; a_t^c]\right) \tag{7}$$

$$r_t = \sigma\left(W_r\, E_w\, y_{t-1} + U_r\, s_{t-1} + A_r\,[a_t^f; a_t^c]\right) \tag{8}$$

$W, W_z, W_r \in \mathbb{R}^{m \times H_d}$, $U, U_z, U_r, A, A_z, A_r \in \mathbb{R}^{H_d \times H_d}$ are learnable parameters of the GRU.

$E_w \in \mathbf{R}^{m \times V}$ is the word embedding matrix, m is the word embedding size and H_d is the size of the decoder hidden state. a_t^f, a_t^c are the outputs of the fact attention and the context attention modules respectively, detailed in the following subsection.
In order to enforce the model to pair output words with words from the textual inputs, we couple the word embedding matrices of both the decoder E_w and the textual context encoder E_c (eq.(3)). We initialize them with GloVe embeddings (Pennington et al., 2014) and allow the network to tune them.
The first hidden state of the decoder $s_0 = [h_f;\ h_c]$ is initialized using a concatenation of the encoded fact (eq.(2)) and the encoded context (eq.(4)) .
At each time step t, after calculating the hidden state of the decoder, the conditional probability distribution over each token y_t of the generated question is computed as the $softmax(W_o\, s_t)$ over all the entries in the output vocabulary, $W_o \in \mathbb{R}^{H_d \times V}$ is the weight matrix of the output layer of the decoder.

3.4 Attention

Our model has two attention modules:
Triple attention over the input triple to determine at each time step t an attention-based encoding of the input fact $a_t^f \in \mathbb{R}^{H_k}$:

$$a_t^f = \alpha_{s,t}\, h_s + \alpha_{p,t}\, h_p + \alpha_{s,t}\, h_o \ , \qquad (9)$$

$\alpha_{s,t}, \alpha_{p,t}, \alpha_{o,t}$ are scalar values calculated by the attention mechanism to determine at each time step which of the encoded subject, predicate, or object the decoder should attend to.
Textual contexts attention over all the hidden states of all the textual contexts $a_t^c \in \mathbb{R}^{H_c}$:

$$a_t^c = \sum_{i=1}^{|C|} \sum_{j=1}^{|c_i|} \alpha_{t,j}^{c_i}\, h_j^{c_i} \ , \qquad (10)$$

$\alpha_{t,j}^{c_i}$ is a scalar value determining the weight of the j^{th} word in the i^{th} context c^i at time step t.

Given a set of encoded input vectors $I = \{h_1, h_2, ...h_k\}$ and the decoder previous hidden state s_{t-1}, the attention mechanism calculates $\alpha_t = \alpha_{i,t}, \ldots, \alpha_{k,t}$ as a vector of scalar weights, each $\alpha_{i,t}$ determines the weight of its corresponding encoded input vector h_i.

$$e_{i,t} = \mathbf{v_a}^\top\ tanh(\mathbf{W_a}\, \mathbf{s_{t-1}} + \mathbf{U_a}\, \mathbf{h_i}) \qquad (11)$$

$$\alpha_{i,t} = \frac{exp\,(e_{i,t})}{\sum_{j=1}^{k} exp\,(e_{j,t})} \ , \qquad (12)$$

where $\mathbf{v_a}, \mathbf{W_a}, \mathbf{U_a}$ are trainable weight matrices of the attention modules. It is important to notice here that we encode each textual context separately using a different GRU, but we calculate an overall attention over all tokens in all textual contexts: at each time step the decoder should ideally attend to only one word from all the input contexts.

What caused the `[C1_NOUN]` of the `[C3_NOUN]` `[S]` ?	
C1	`[S]` **`death`** `by [O]`
	`[S]` **`[C1_NOUN]`** `[C1_ADP] [O]`
C2	Disease
	`[C2_NOUN]`
C3	Musical **artist**
	`[C3_ADJ]` **`[C3_NOUN]`**

Table 1: An annotated example of part-of-speech copy actions from several input textual contexts (C1, C2, C3), the words or placeholders in bold are copied in the generated question

3.5 Part-Of-Speech Copy Actions

We use the method of (Luong et al., 2015) by modeling all the copy actions on the data level through an annotation scheme. This method treats the model as a black box, which makes it adaptable to any text generation model. Instead of using positional copy actions, we use the part-of-speech information to decide the alignment process between the input and output texts to the model. Each word in every input textual context is replaced by a special token containing a combination of its context id (e.g. `C1`) and its POS tag (e.g. `NOUN`). Then, if a word in the output question matches a word in a textual context, it is replaced with its corresponding tag as shown in Table 1.
Unlike (Serban et al., 2016; Lebret et al., 2016) we model the copy actions in the input and the output levels. Our model does not have the drawback of losing the semantic information when replacing words with generic placeholders, since we provide the model with the input triple through the fact encoder. During inference the model chooses to either output words from the vocabulary or special tokens to copy from the textual contexts. In

a post-processing step those special tokens are replaced with their original words from the textual contexts.

4 Textual contexts dataset

As a source of question paired with KB triples we use the SimpleQuestions dataset (Bordes et al., 2015). It consists of 100K questions with their corresponding triples from Freebase, and was created manually through crowdsourcing. When asked to form a question from an input triple, human annotators usually tend to mainly focus on expressing the predicate of the input triple. For example, given a triple with the predicate `fb:spacecraft/manufacturer` the user may ask *"What is the manufacturer of `[S]` ?"*. Annotators may specify the entity type of the subject or the object of the triple: *"What is the manufacturer of the **spacecraft** `[S]`?"* or *"Which **company** manufactures `[S]`?"*. Motivated by this example we chose to associate each input triple with three textual contexts of three different types. The first is a phrase containing lexicalization of the predicate of the triple. The second and the third are two phrases containing the entity type of the subject and the object of the triple. In what follows we show the process of collection and preprocessing of those textual contexts.

4.1 Collection of Textual Contexts

We extend the set of triples given in the SimpleQuestions dataset by using the FB5M (Bordes et al., 2015) subset of Freebase. As a source of text documents, we rely on Wikipedia articles.

Predicate textual contexts: In order to collect textual contexts associated with the SimpleQuestions triples, we follow the distant supervision setup for relation extraction (Mintz et al., 2009). The distant supervision assumption has been effective in creating training data for relation extraction and shown to be 87% correct (Riedel et al., 2010) on Wikipedia text.
First, we align each triple in the FB5M KB to sentences in Wikipedia if the subject and the object of this triple co-occur in the same sentence. We use a simple string matching heuristic to find entity mentions in text[2]. Afterwards we reduce the sentence to the set of words that appear on the dependency path between the subject and the object mentions in the sentence. We replace the positions of the subject and the object mentions with `[S]` and `[O]` to the keep track of the information about the direction of the relation. The top occurring pattern for each predicate is associated to this predicate as its textual context. Table 2 shows examples of predicates and their corresponding textual context.

Freebase Relation	Predicate Textual Context
person/place_of_birth	[O] is birthplace of [S]
currency/former_countries	[S] was currency of [O]
dish/cuisine	[O] dish [S]
airliner_accident/flight_origin	[S] was flight from [O]
film_featured_song/performer	[S] is release by [O]
airline_accident/operator	[S] was accident for [O]
genre/artists	[S] became a genre of [O]
risk_factor/diseases	[S] increases likelihood of [O]
book/illustrations_by	[S] illustrated by [O]
religious_text/religion	[S] contains principles of [O]
spacecraft/manufacturer	[S] spacecraft developed by [O]

Table 2: Table showing an example of textual contexts extracted for freebase predicates

Sub-Type and Obj-Type textual contexts: We use the labels of the entity types as the sub-type and obj-type textual contexts. We collect the list of entity types of each entity in the FB5M through the predicate `fb:type/instance`. If an entity has multiple entity types we pick the entity type that is mentioned the most in the first sentence of each Wikipedia article. Thus the textual contexts will opt for entity types that is more natural to appear in free text and therefore questions.

4.2 Generation of Special tokens

To generate the special tokens for copy actions (sec. 3.5) we run POS tagging on each of the input textual contexts[3]. We replace every word in each textual context with a combination of its context id (e.g. `C1`) and its POS tag (e.g. `NOUN`). If the same POS tag appears multiple times in the textual context, it is given an additional id (e.g. `C1_NOUN_2`). If a word in the output question overlaps with a word in the input textual context, this word is replaced by its corresponding tag.
For sentence and word tokenization we use the Regex tokenizer from the NLTK toolkit (Bird, 2006), and for POS tagging and dependency pars-

[2] We map Freebase entities to Wikidata through the Wikidata property P646, then we extract their labels and aliases. We use the Wikidata truthy dump: `https://dumps.wikimedia.org/wikidatawiki/entities/`

[3] For the predicate textual contexts we run pos tagging on the original text not the lexicalized dependency path

		Train	Valid	Test
pred	# pred	169.4	24.2	48.4
	# samples	55566.7	7938.1	15876.2
	% samples	70.0 ± 2.77	10.0 ± 1.236	20.0 ± 2.12
sub-types	# types	112.7	16.1	32.2
	# samples	60002.6	8571.8	17143.6
	% samples	70.0 ± 7.9	10.0 ± 3.6	20.0 ± 6.2
obj-types	# types	521.6	189.9	282.2
	# samples	57878.1	8268.3	16536.6
	% samples	70.0 ± 4.7	10.0 ± 2.5	20.0 ± 3.8

Table 3: Dataset statistics across 10 folds for each experiment

ing we use the Spacy[4] implementation.

5 Experiments

5.1 Zero-Shot Setups

We develop three setups that follow the same procedure as (Levy et al., 2017) for Zero-Shot relation extraction to evaluate how our model generalizes to: 1) unseen predicates, 2) unseen sub-types and 3) unseen obj-types.

For the unseen predicates setup we group all the samples in SimpleQuestions by the predicate of the input triple, and keep groups that contain at least 50 samples. Afterwards we randomly split those groups to 70% train, 10% valid and 20% test mutual exclusive sets respectively. This guarantees that if the predicate `fb:person/place_of_birth` for example shows during test time, the training and validation set will not contain any input triples having this predicate. We repeat this process to create 10 cross validation folds, in our evaluation we report the mean and standard deviation results across those 10 folds. While doing this we make sure that the number of samples in each fold – not only unique predicates – follow the same 70%, 30%, 10% distribution. We repeat the same process for the subject entity types and object entity types (answer types) individually. Similarly, for example in the unseen object-type setup, the question *"Which **artist** was born in Berlin?"* appearing in the test set means that, there is no question in the training set having an entity of type ***artist***. Table 3 shows the mean number of samples, predicates, sub-types and obj-types across the 10 folds for each experiment setup.

[4]https://spacy.io/

5.2 Baselines

`SELECT` is a baseline built from (Serban et al., 2016) and adapted for the zero shot setup. During test time given a fact F, this baseline picks a fact F_c from the training set and outputs the question that corresponds to it. For evaluating unseen predicates, F_c has the same answer type (obj-type) as F. And while evaluating unseen sub-types or obj-types, F_c and F have the same predicate.

`R-TRANSE` is an extension that we propose for `SELECT`. The input triple is encoded using the concatenation of the TransE embeddings of the subject, predicate and object. At test time, `R-TRANSE` picks a fact from the training set that is the closest to the input fact using cosine similarity and outputs the question that corresponds to it. We provide two versions of this baseline: **`R-TRANSE`** which indexes and retrieves raw questions with only a single placeholder for the subject label, such as in (Serban et al., 2016). And **`R-TRANSE`**$_{\texttt{copy}}$ which indexes and retrieves questions using our copy actions mechanism (sec. 3.5).

`IR` is an information retrieval baseline. Information retrieval has been used before as baseline for QG from text input (Rush et al., 2015; Du et al., 2017). We rely on the textual context of each input triple as the search keyword for retrieval. First, the IR baseline encodes each question in the training set as a vector of TF-IDF weights (Joachims, 1997) and then does dimensionality reduction through LSA (Halko et al., 2011). At test time the textual context of the input triple is converted into a dense vector using the same process and then the question with the closest cosine distance to the input is retrieved. We provide two versions of this baseline: **`IR`** on raw text and **`IR`**$_{\texttt{copy}}$ on text with our placeholders for copy actions.

`Encoder-Decoder.` Finally, we compare our model to the Encoder-Decoder model with a single placeholder, the best performing model from (Serban et al., 2016). We initialize the encoder with TransE embeddings and the decoder with GloVe word embeddings. Although this model was not originally built to generalize to unseen predicates and entity types, it has some generalization abilities represented in the encoded infor-

mation in the pre-trained embeddings. Pretrained KB terms and word embeddings encode relations between entities or between words as translations in the vector space. Thus the model might be able to map new classes or predicates in the input fact to new words in the output question.

5.3 Training & Implementation Details

To train the neural network models we optimize the negative log-likelihood of the training data with respect to all the model parameters. For that we use the RMSProp optimization algorithm with a decreasing learning rate of 0.001, mini-batch size $= 200$, and clipping gradients with norms larger than 0.1. We use the same vocabulary for both the textual context encoders and the decoder outputs. We limit our vocabulary to the top $30,000$ words including the special tokens. For the word embeddings we chose GloVe (Pennington et al., 2014) pretrained embeddings of size 100. We train TransE embeddings of size $H_k = 200$, on the FB5M dataset (Bordes et al., 2015) using the TransE model implementation from (Lin et al., 2015). We set GRU hidden size of the decoder to $H_d = 500$, and textual encoder to $H_c = 200$. The networks hyperparameters are set with respect to the final BLEU-4 score over the validation set. All neural networks are implemented using Tensorflow (Abadi et al., 2015). All experiments and models source code are publicly available[5] for the sake of reproducibility.

5.4 Automatic Evaluation Metrics

To evaluate the quality of the generated question, we compare the original labeled questions by human annotators to the ones generated by each variation of our model and the baselines. We rely on a set of well established evaluation metrics for text generation: BLEU-1, BLEU-2, BLEU-3, BLEU-4 (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014) and ROUGE_L (Lin, 2004).

5.5 Human Evaluation

Automatic Metrics for evaluating text generation such as BLEU and METEOR give an measure of how close the generated questions are to the target correct labels. However, they still suffer from many limitations (Novikova et al., 2017). Automatic metrics might not be able to evaluate directly whether a specific predicate was explicitly mentioned in the generated text or not.
As an example, taking a target question and two corresponding generated questions A and B:

	What kind of film is kill bill vol. 2?	**BLEU**
A)	**What is *the name of* the film kill bill vol. 2?**	71
B)	**Which genre is kill bill vol. 2 in?**	55

We can find that the sentence A having a better BLEU score than B although it is not able to express the correct target predicate (*film genre*). For that reason we decide to run two further human evaluations to directly measure the following:
Predicate identification: annotators were asked to indicate whether the generated question contains the given predicate in the fact or not, either directly or implicitly.
Naturalness: following (Ngomo et al., 2013), we measure the comprehensibility and readability of the generated questions. Each annotator was asked to rate each generated question using a scale from 1 to 5, where: (5) perfectly clear and natural, (3) artificial but understandable, and (1) completely not understandable. We run our studies on 100 randomly sampled input facts alongside with their corresponding generated questions by each of the systems using the help of 4 annotators.

6 Results & Discussion

Automatic Evaluation Table 4 shows results of our model compared to all other baselines across all evaluation metrics. Our that encodes the KB fact and textual contexts achieves a significant enhancement over all the baselines in all evaluation metrics, with $+\mathbf{2.04}$ BLEU-4 score than the Encoder-Decoder baseline. Incorporating the part-of-speech copy actions further improves this enhancement to reach $+\mathbf{2.39}$ BLEU-4 points.
Among all baselines, the Encoder-Decoder baseline and the R-TRANSE baseline performed the best. This shows that TransE embeddings encode intra-predicates information and intra-class-types information to a great extent, and can generalize to some extent to unseen predicates and class types.

Similar patterns can be seen in the evaluation on unseen sub-types and obj-types (Table 5). Our model with copy actions was able to outperform

[5] https://github.com/hadyelsahar/Zeroshot-QuestionGeneration

	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE_L	METEOR
Unseen Predicates	SELECT	46.81 ± 2.12	38.62 ± 1.78	31.26 ± 1.9	23.66 ± 2.22	52.04 ± 1.43	27.11 ± 0.74
	IR	48.43 ± 1.64	39.13 ± 1.34	31.4 ± 1.66	23.59 ± 2.36	52.88 ± 1.24	27.34 ± 0.55
	IR_COPY	48.22 ± 1.84	38.82 ± 1.5	31.01 ± 1.72	23.12 ± 2.24	52.72 ± 1.26	27.24 ± 0.57
	R-TRANSE	49.09 ± 1.69	40.75 ± 1.42	33.4 ± 1.7	25.97 ± 2.22	54.07 ± 1.31	28.13 ± 0.54
	$\text{R-TRANSE}_\text{COPY}$	49.0 ± 1.76	40.63 ± 1.48	33.28 ± 1.74	25.87 ± 2.23	54.09 ± 1.35	28.12 ± 0.57
	Encoder-Decoder	58.92 ± 2.05	47.7 ± 1.62	38.18 ± 1.86	28.71 ± 2.35	59.12 ± 1.16	34.28 ± 0.54
	Our-Model	60.8 ± 1.52	49.8 ± 1.37	40.32 ± 1.92	30.76 ± 2.7	60.07 ± 0.9	35.34 ± 0.43
	$\text{Our-Model}_\text{copy}$	**62.44** ± 1.85	**50.62** ± 1.46	**40.82** ± 1.77	**31.1** ± 2.46	**61.23** ± 1.2	**36.24** ± 0.65

Table 4: Evaluation results of our model and all other baselines for the unseen predicate evaluation setup

	Model	BLEU-4	ROUGE_L
Sub-Types	R-TRANSE	32.41 ± 1.74	59.27 ± 0.92
	Encoder-Decoder	42.14 ± 2.05	68.95 ± 0.86
	Our-Model	42.13 ± 1.88	69.35 ± 0.9
	$\text{Our-Model}_\text{copy}$	**42.2** ± 2.0	**69.37** ± 1.0
Obj-Types	R-TRANSE	30.59 ± 1.3	57.37 ± 1.17
	Encoder-Decoder	37.79 ± 2.65	65.69 ± 2.25
	Our-Model	37.78 ± 2.02	65.51 ± 1.56
	$\text{Our-Model}_\text{copy}$	**38.02** ± 1.9	**66.24** ± 1.38

Table 5: Automatic evaluation of our model against selected baselines for unseen sub-types and obj-types

Model	% Pred. Identified	Natural.
Encoder-Decoder	6	3.14
Our-Model (No Copy)	6	2.72
$\text{Our-Model}_\text{copy}$ (Types context)	**37**	**3.21**
$\text{Our-Model}_\text{copy}$ (All contexts)	**46**	2.61

Table 6: results of Human evaluation on % of predicates identified and naturalness 0-5

all the other systems. Majority of systems have reported a significantly higher BLEU-4 scores in these two tasks than when generalizing to unseen predicates (+12 and +8 BLEU-4 points respectively). This indicates that these tasks are relatively easier and hence our models achieve relatively smaller enhancements over the baselines.

Human Evaluation Table 6 shows how different variations of our system can express the unseen predicate in the target question with comparison to the Encoder-Decoder baseline.
Our proposed copy actions have scored a significant enhancement in the identification of unseen predicates with up to +40% more than best performing baseline and our model version without the copy actions.

By examining some of the generated questions (Table 7) we see that models without copy actions can generalize to unseen predicates that only have a very similar freebase predicate in the training set. For example `fb:tv_program/language` and `fb:film/language`, if one of those predicates exists in the training set the model can use the same questions for the other during test time.
Copy actions from the sub-type and the obj-type textual contexts can generalize to a great extent to unseen predicates because of the overlap between the predicate and the object type in many questions (Example 2 Table 7). Adding the predicate context to our model has enhanced model performance for expressing unseen predicates by +9% (Table 6). However we can see that it has affected the naturalness of the question. The post processing step does not take into consideration that some verbs and prepositions do not fit in the sentence structure, or that some words are already existing in the question words (Example 4 Table 7). This does not happen as much when having copy actions from the sub-type and the obj-type contexts because they are mainly formed of nouns which are more interchangeable than verbs or prepositions. A post-processing step to reform the question instead of direct copying from the input source is considered in our future work.

7 Conclusion

In this paper we presented a new neural model for question generation from knowledge bases, with a main focus on predicates, subject types or object types that were not seen at the training phase (Zero-Shot Question Generation). Our model is based on an encoder-decoder architecture that leverages textual contexts of triples, two attention layers for triples and textual contexts and

1	**Reference**	**what language is spoken in the tv show three sheets?**
	Enc-Dec.	in what **language** is three sheets in?
	Our-Model	what the the player is the three sheets?
	Our-Model$_{\text{Copy}}$	what is the **language** of three sheets?
2	**Reference**	**how is roosevelt in Africa classified?**
	Enc-Dec.	what is the name of a roosevelt in Africa?
	Our-Model	what is the name of the movie roosevelt in Africa?
	Our-Model$_{\text{Copy}}$	what is a **genre** of roosevelt in Africa?
3	**Reference**	**where can 5260 philvron be found**?
	Enc-Dec.	what is a release some that 5260 philvron wrote?
	Our-Model	what is the name of an artist 5260 philvron?
	Our-Model$_{\text{Copy}}$	which **star system** contains the star system body 5260 philvron?
4	**Reference**	**which university did ezra cornell create**?
	Enc-Dec.	which films are part of ezra cornell?
	Our-Model	what is a position of ezra cornell?
	Our-Model$_{\text{Copy}}$	what *founded* the name of a university that ezra cornell **founded**?
5	**Reference**	**who founded snocap , inc .**?
	Enc-Dec.	which asian snocap is most as?
	Our model	what is the name of a person of snocap?
	Our-Model$_{\text{Copy}}$	who is the **person behind** snocap?

Table 7: Examples of generated questions from different systems in comparison

finally a part-of-speech copy action mechanism. Our method exhibits significantly better results for Zero-Shot QG than a set of strong baselines including the state-of-the-art question generation from KB. Additionally, a complimentary human evaluation, helps in showing that the improvement brought by our part-of-speech copy action mechanism is even more significant than what the automatic evaluation suggests. The source code and the collected textual contexts are provided for the community [6]

[6] https://github.com/hadyelsahar/Zeroshot-QuestionGeneration

Acknowledgements

This research is partially supported by the Answering Questions using Web Data (WDAqua) project, a Marie Skłodowska-Curie Innovative Training Network under grant agreement No 642795, part of the Horizon 2020 programme.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. https://www.tensorflow.org/.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473. http://arxiv.org/abs/1409.0473.

Steven Bird. 2006. NLTK: the natural language toolkit. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*. http://aclweb.org/anthology/P06-4018.

Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *CoRR* abs/1506.02075. http://arxiv.org/abs/1506.02075.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR* abs/1406.1078.

Michael J. Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*. pages 376–380. http://aclweb.org/anthology/W/W14/W14-3348.pdf.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. pages 875–886. https://aclanthology.info/papers/D17-1091/d17-1091.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. pages 1342–1352. https://doi.org/10.18653/v1/P17-1123.

Martha Evens and Joel Michael. 2006. One-on-one tutoring by humans and machines. *Computer Science Department, Illinois Institute of Technology* .

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1535–1545. http://www.aclweb.org/anthology/D11-1142.

Arthur C Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M Louwerse. 2004. Autotutor: A tutor with dialogue in natural language. *Behavior Research Methods* 36(2):180–192.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1154.pdf.

Çaglar Gülçehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1014.pdf.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review* 53(2):217–288. https://doi.org/10.1137/090771806.

Michael Heilman and Noah A. Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA*. pages 609–617. http://www.aclweb.org/anthology/N10-1086.

Thorsten Joachims. 1997. A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997*. pages 143–151.

Mitesh M. Khapra, Dinesh Raghu, Sachindra Joshi, and Sathish Reddy. 2017. Generating natural language question-answer pairs from a knowledge graph using a RNN based question generation model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*. pages 376–385. https://aclanthology.info/pdf/E/E17/E17-1036.pdf.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR* abs/1411.2539. http://arxiv.org/abs/1411.2539.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. pages 1203–1213. http://aclweb.org/anthology/D/D16/D16-1128.pdf.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*. pages 333–342. https://doi.org/10.18653/v1/K17-1034.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*. Barcelona, Spain, volume 8.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*. pages 2181–2187. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9571.

Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine trans-

lation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. pages 11–19. http://aclweb.org/anthology/P/P15/P15-1002.pdf.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*. pages 1003–1011. http://www.aclweb.org/anthology/P09-1113.

Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, Christina Unger, Jens Lehmann, and Daniel Gerber. 2013. Sorry, i don't speak SPARQL: translating SPARQL queries into natural language. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*. pages 977–988. http://dl.acm.org/citation.cfm?id=2488473.

Jekaterina Novikova, Ondrej Dusek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. pages 2241–2252. https://aclanthology.info/papers/D17-1238/d17-1238.

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, July 6-12, 2002, Philadelphia, PA, USA.*. pages 311–318. http://www.aclweb.org/anthology/P02-1040.pdf.

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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1532–1543. http://aclweb.org/anthology/D/D14/D14-1162.pdf.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III*. pages 148–163. https://doi.org/10.1007/978-3-642-15939-8_10.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. pages 379–389. http://aclweb.org/anthology/D/D15/D15-1044.pdf.

Iulian Vlad Serban, Alberto García-Durán, Çaglar Gülçehre, Sungjin Ahn, Sarath Chandar, Aaron C. Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1056.pdf.

Dominic Seyler, Mohamed Yahya, and Klaus Berberich. 2015. Generating quiz questions from knowledge graphs. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*. pages 113–114. https://doi.org/10.1145/2740908.2742722.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. pages 1577–1586. http://aclweb.org/anthology/P/P15/P15-1152.pdf.

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*. pages 935–943.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. pages 3104–3112.