Generating Factoid Questions With Recurrent Neural Networks: The 30M Factoid Question-Answer Corpus

Iulian Vlad Serban[*◦]
University of Montreal
2920 chemin de la Tour,
Montréal, QC, Canada

Alberto García-Durán[*⋄]
Université de Technologie de Compiègne CNRS
Rue du Dr Schweitzer,
Compigne, France

Caglar Gulcehre[◦]
University of Montreal
2920 chemin de la Tour,
Montréal, QC, Canada

Sungjin Ahn[◦]
University of Montreal
2920 chemin de la Tour,
Montréal, QC, Canada

Sarath Chandar[◦]
University of Montreal
2920 chemin de la Tour,
Montréal, QC, Canada

Aaron Courville[◦]
University of Montreal
2920 chemin de la Tour,
Montréal, QC, Canada

Yoshua Bengio[†◦]
University of Montreal
2920 chemin de la Tour,
Montréal, QC, Canada

Abstract

Over the past decade, large-scale supervised learning corpora have enabled machine learning researchers to make substantial advances. However, to this date, there are no large-scale question-answer corpora available. In this paper we present the 30M Factoid Question-Answer Corpus, an enormous question-answer pair corpus produced by applying a novel neural network architecture on the knowledge base Freebase to transduce facts into natural language questions. The produced question-answer pairs are evaluated both by human evaluators and using automatic evaluation metrics, including well-established machine translation and sentence similarity metrics. Across all evaluation criteria the question-generation model outperforms the competing template-based baseline. Furthermore, when presented to human evaluators, the generated questions appear to be comparable in quality to real human-generated questions.

1 Introduction

A major obstacle for training question-answering (QA) systems has been due to the lack of labeled data. The question answering field has focused on building QA systems based on traditional information retrieval procedures (Lopez et al., 2011; Dumais et al., 2002; Voorhees and Tice, 2000). More recently, researchers have started to utilize large-scale knowledge bases (KBs) (Lopez et al., 2011), such as Freebase (Bollacker et al., 2008), WikiData (Vrandečić and Krötzsch, 2014) and Cyc (Lenat and Guha, 1989).[1] Bootstrapping QA systems with such structured knowledge is clearly beneficial, but it is unlikely alone to overcome the lack of labeled data. To take into account the rich and complex nature of human language, such as paraphrases and ambiguity, it would appear that labeled question and answer pairs are necessary. The need for such labeled pairs is even more critical for training neural network-based QA systems, where researchers until now have relied mainly on hand-crafted rules and heuristics to synthesize artificial QA corpora (Bordes et al., 2014; Bordes et al., 2015).

Motivated by these recent developments, in this paper we focus on generating questions based on the Freebase KB. We frame question generation as a transduction problem starting from a Freebase fact, represented by a triple consisting of a subject, a relationship and an object, which is trans-

* First authors.
◦ Email: {iulian.vlad.serban,caglar.gulcehre, sungjin.ahn,sarath.chandar.anbil.parthipan, aaron.courville,yoshua.bengio}@umontreal.ca
⋄ Email: alberto.garcia-duran@utc.fr
† CIFAR Senior Fellow

[1] Freebase is now a part of WikiData.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 588–598,
Berlin, Germany, August 7-12, 2016.

duced into a question about the subject, where the object is the correct answer (Bordes et al., 2015). We propose several models, largely inspired by recent neural machine translation models (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015), and we use an approach similar to Luong et al. (2015) for dealing with the problem of rare-words. We evaluate the produced questions in a human-based experiment as well as with respect to automatic evaluation metrics, including the well-established machine translation metrics BLEU and METEOR and a sentence similarity metric. We find that the question-generation model outperforms the competing template-based baseline, and, when presented to untrained human evaluators, the produced questions appear to be indistinguishable from real human-generated questions. This suggests that the produced question-answer pairs are of high quality and therefore that they will be useful for training QA systems. Finally, we use the best performing model to construct a new factoid question-answer corpus – The 30M Factoid Question-Answer Corpus – which is made freely available to the research community.[2]

2 Related Work

Question generation has attracted interest in recent years with notable work by Rus et al. (2010), followed by the increasing interest from the Natural Language Generation (NLG) community. A simple rule-based approach was proposed in different studies as *wh-fronting* or *wh-inversion* (Kalady et al., 2010; Ali et al., 2010). This comes at the disadvantage of not making use of the semantic content of words apart from their syntactic role. The problem of determining the *question type* (e.g. that a *Where-question* should be triggered for locations), which requires knowledge of the category type of the elements involved in the sentence, has been addressed in two different ways: by using named entity recognizers (Mannem et al., 2010; Yao and Zhang, 2010) or semantic role labelers (Chen et al., 2009). In Curto et al. (2012) questions are split into classes according to their syntactic structure, prefix of the question and the category of the answer, and then a pattern is learned to generate questions for that class of questions. After the identification of key points, Chen et al. (2009) apply handcrafted-templates to generate questions framed in the right target expression by following the analysis of Graesser et al. (1992), who classify questions according to a taxonomy consisting of 18 categories.

The works discussed so far propose ways to map unstructured text to questions. This implies a two-step process: first, transform a text into a symbolic representation (e.g. a syntactic representation of the sentence), and second, transform the symbolic representation of the text into the question (Yao et al., 2012). On the other hand, going from a symbolic representation (structured information) to a question, as we will describe in the next section, only involves the second step. Closer to our approach is the work by Olney et al. (2012). They take triples as input, where the edge relation defines the question template and the head of the triple replaces the placeholder token in the selected question template. In the same spirit, Duma et al. (2013) generate short descriptions from triples by using templates defined by the relationship and replacing accordingly the placeholder tokens for the subject and object.

Our baseline is similar to that of Olney et al. (2012), where a set of relationship-specific templates are defined. These templates include placeholders to replace the string of the subject. The main difference with respect to their work is that our baseline does not explicitly define these templates. Instead, each relationship has as many templates as there are different ways of framing a question with that relationship in the training set. This yields more diverse and semantically richer questions by effectively taking advantage of the fact-question pairs, which Olney et al. did not have access to in their experiments.

Unlike the work by Berant and Liang (2014), which addresses the problem of deterministically generating a set of candidate logical forms with a canonical realization in natural language for each, our work addresses the inverse problem: given a logical form (fact) it outputs the associated question.

It should also be noted that recent work in question answering have used simpler rule-based and template-based approaches to generate synthetic questions to address the lack of question-answer pairs to train their models (Bordes et al., 2014; Bordes et al., 2015).

[2] www.agarciaduran.org

3 Task Definition

3.1 Knowledge Bases

In general, a KB can be viewed as a multi-relational graph, which consists of a set of nodes (entities) and a set of edges (relationships) linking nodes together. In Freebase (Bollacker et al., 2008) these relationships are directed and always connect exactly two entities. For example, in Freebase the two entities *fires_creek* and *nantahala_national_forest* are linked together by the relationship *contained_by*. Since the triple $\{$*fires_creek*, *contained_by*, *nantahala_national_forest*$\}$ represents a complete and self-contained piece of information, it is also called a *fact* where *fires_creek* is the subject (head of the edge), *contained_by* is the relationship and *nantahala_national_forest* is the object (tail of the edge).

3.2 Transducing Facts to Questions

We aim to transduce a fact into a question, such that:

1. The question is concerned with the subject and relationship of the fact, and
2. The object of the fact represents a valid answer to the generated question.

We model this in a probabilistic framework as a directed graphical model:

$$P(Q|F) = \prod_{n=1}^{N} P(w_n|w_{<n}, F), \qquad (1)$$

where $F = (subject, relationship, object)$ represents the fact, $Q = (w_1, \ldots, w_N)$ represents the question as a sequence of tokens $w_1, \ldots, w_N$, and $w_{<n}$ represents all the tokens generated before token w_n. In particular, w_N represents the question mark symbol '?'.

3.3 Dataset

We use the SimpleQuestions dataset (Bordes et al., 2015) in order to train our models. This is by far the largest dataset of question-answer pairs created by humans based on a KB. It contains over 100K question-answer pairs created by users on Amazon Mechanical Turk[3] in English based on the Freebase KB. In order to create the questions, human participants were shown one whole Freebase fact at a time and they were asked to phrase a question such that the object of the presented fact becomes the answer of the question.[4] Consequently, both the subject and the relationship are explicitly given in each question. But indirectly characteristics of the object may also be given since the humans have an access to it as well. Often when phrasing a question the annotators tend to be more informative about the target object by giving specific information about it in the question produced. For example, in the question *What city is the American actress X from?* the city name given in the object informs the human participant that it was in America - information, which was not provided by either the subject or relationship of the fact. We have also observed that the questions are often ambiguous: that is, one can easily come up with several possible answers that may fit the specifications of the question. Table 1 shows statistics of the dataset.

Questions	Entities	Relationships	Words
108,442	131,684	1,837	∼77k

Table 1: Statistics of SimpleQuestions

4 Model

We propose to attack the problem with the models inspired by the recent success of neural machine translation models (Sutskever et al., 2014; Bahdanau et al., 2015). Intuitively, one can think of the transduction task as a "lossy translation" from structured knowledge (facts) to human language (questions in natural language), where certain aspects of the structured knowledge is intentionally left out (e.g. the name of the object). These models typically consist of two components: an encoder, which encodes the source phrase into one or several fixed-size vectors, and a decoder, which decodes the target phrase based on the results of the encoder.

4.1 Encoder

In contrast to the neural machine translation framework, our source language is not a proper language but instead a sequence of three variables making up a fact. We propose an encoder sub-model, which encodes each atom of the fact into an embedding. Each atom $\{s, r, o\}$, may

[3] `www.mturk.com`

[4] It is not necessary for the object to be the only answer, but it is required to be one of the possible answers.

stand for subject, relationship and object, respectively, of a fact $F = (s, r, o)$ is represented as a 1-of-K vector x_{atom}, whose embedding is obtained as $e_{\text{atom}} = E_{\text{in}} x_{\text{atom}}$, where $E_{\text{in}} \in \mathbb{R}^{D_{\text{Enc}} \times K}$ is the embedding matrix of the input vocabulary and K is the size of that vocabulary. The encoder transforms this embedding into $\text{Enc}(F)_{\text{atom}} \in \mathbb{R}^{H_{\text{Dec}}}$ as $\text{Enc}(F)_{\text{atom}} = W_{\text{Enc}} e_{\text{atom}}$, where $W_{\text{Enc}} \in \mathbb{R}^{H_{\text{Dec}} \times D_{\text{Enc}}}$.

This embedding matrix, E_{in}, could be another parameter of the model to be learned, however, as discussed later (see Section 4.3), we have learned it separately and beforehand with *TransE* (Bordes et al., 2013), a model aimed at modeling this kind of multi-relational data. We fix it and do not allow the encoder to tune it during training.

We call *fact embedding* $\text{Enc}(F) \in \mathbb{R}^{3H_{\text{Dec}}}$ the concatenation $[\text{Enc}(F)_s,\ \text{Enc}(F)_r,\ \text{Enc}(F)_o]$ of the atom embeddings, which is the input for the next module.

4.2 Decoder

For the decoder, we use a GRU recurrent neural network (RNN) (Cho et al., 2014) with an attention-mechanism (Bahdanau et al., 2015) on the encoder representation to generate the associated question Q to that fact F. Recently, it has been shown that the GRU RNN performs equally well across a range of tasks compared to other RNN architectures, such as the LSTM RNN (Greff et al., 2015). The hidden state of the decoder RNN is computed at each time step n as:

$$g_n^r = \sigma(W_r E_{out} w_{n-1} + C_r c(F, h_{n-1}) + U_r h_{n-1}) \quad (2)$$

$$g_n^u = \sigma(W_u E_{out} w_{n-1} + C_u c(F, h_{n-1}) + U_u h_{n-1}) \quad (3)$$

$$\tilde{h} = \tanh(W E_{out} w_{n-1} + C c(F, h_{n-1}) + U(g_n^r \circ h_{n-1})) \quad (4)$$

$$h_n = g_n^u \circ h_{n-1} + (1 - g_n^u) \circ \tilde{h}, \quad (5)$$

where σ is the sigmoid function, s.t. $\sigma(x) \in [0, 1]$, and the circle, $\circ$, represents element-wise multiplication. The initial state h_0 of this RNN is given by the output of a feedforward neural network fed with the fact embedding. The product $E_{out} w_n \in \mathbb{R}^{D_{\text{Dec}}}$ is the decoder embedding vector corresponding to the word w_n (coded as a 1-of-V vector, with V being the size of the output vocabulary), the variables $U_r, U_u, U, C_r, C_u, C \in \mathbb{R}^{H_{\text{Dec}} \times H_{\text{Dec}}}$, $W_r, W_u, W \in \mathbb{R}^{H_{\text{Dec}} \times D_{\text{Dec}}}$ are the parameters of the GRU and $c(F, h_{n-1})$ is the context vector (defined below Eq. 6). The vector g^r is called the *reset gate*, g^u as the *update gate* and $\tilde{h}$ the *candidate activation*. By adjusting g^r and g^u appropriately, the model is able to create linear *skip-connections* between distant hidden states, which in turn makes the credit assignment problem easier and the gradient signal stronger to earlier hidden states. Then, at each time step n the set of probabilities over word tokens is given by applying a softmax layer over $V_o \tanh(V_h h_n + V_w E_{out} w_{n-1} + V_c c(F, h_{n-1}))$, where $V_o \in \mathbb{R}^{V \times H_{\text{Dec}}}$, $V_h, V_c \in \mathbb{R}^{H_{\text{Dec}} \times H_{\text{Dec}}}$ and $V_w \in \mathbb{R}^{H_{\text{Dec}} \times D_{\text{Dec}}}$. Lastly, the function $c(F, h_{n-1})$ is computed using an attention-mechanism:

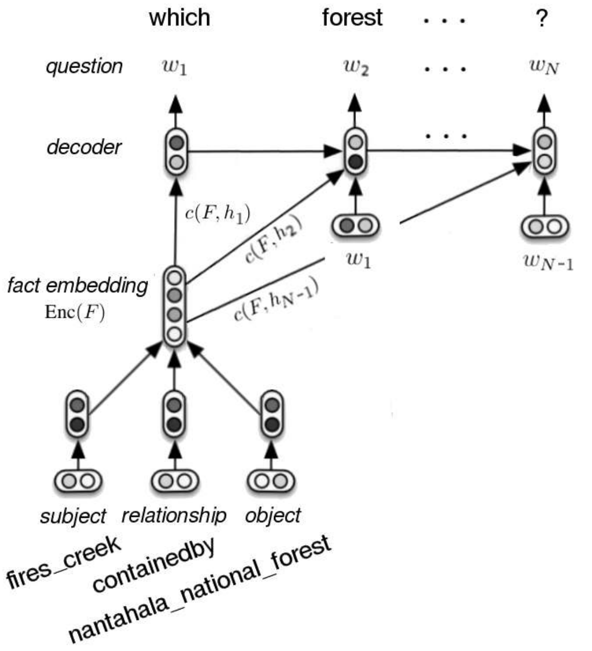

Figure 1: The computational graph of the question-generation model, where $\text{Enc}(F)$ is the fact embedding produced by the encoder model, and $c(F, h_{n-1})$ for $n = 1, \ldots, N$ is the fact representation weighed according to the attention-mechanism, which depends on both the fact F and the previous hidden state of the decoder RNN h_{n-1} . For the sake of simplicity, the attention-mechanism is not shown explicitly.

$$c(F, h_{n-1}) = \alpha_{\text{s},n-1} \text{Enc}(F)_{\text{s}} + \alpha_{\text{r},n-1} \text{Enc}(F)_{\text{r}} + \alpha_{\text{o},n-1} \text{Enc}(F)_{\text{o}}, \quad (6)$$

where $\alpha_{\text{s},n-1}, \alpha_{\text{r},n-1}, \alpha_{\text{r},n-1}$ are real-valued scalars, which weigh the contribution of the subject, relationship and object representations.

They correspond to the *attention* of the model, and are computed by applying a one-layer neural network with tanh-activation function on the encoder representations of the fact, $\text{Enc}(F)$, and the previous hidden state of the RNN, h_{n-1}, followed by the sigmoid function to restrict the attention values to be between zero and one. The need for the attention-mechanism is motivated by the intuition that the model needs to attend to the subject only once during the generation process while attending to the relationship at all other times during the generation process. The model is illustrated in Figure 1.

4.3 Modeling the Source Language

A particular problem with the model presented above is related to the embeddings for the entities, relationships and tokens, which all have to be learned in one way or another. If we learn these naively on the SimpleQuestions training set, the model will perform poorly when it encounters previously unseen entities, relationships or tokens. Furthermore, the multi-relational graph defined by the facts in SimpleQuestions is extremely sparse, i.e. each node has very few edges to other nodes, as can be expected due to high ratio of unique entities over number of examples. Therefore, even for many of the entities in SimpleQuestions, the model may perform poorly if the embedding is learned solely based on the SimpleQuestions dataset alone.

On the source side, we can resolve this issue by initializing the subject, relationship and object embeddings to those learned by applying multi-relational embedding-based models to the knowledge base. Multi-relational embedding-based models (Bordes et al., 2011) have recently become popular to learn distributed vector embeddings for knowledge bases, and have shown to scale well and yield good performance. Due to its simplicity and good performance, we choose to use TransE (Bordes et al., 2013) to learn such embeddings. TransE is a translation-based model, whose energy function is trained to output low values when the fact expresses true information, i.e. a fact which exists in the knowledge base, and otherwise high values. Formally, the energy function is defined as $f(s, r, o) = ||e_s + e_r - e_o||_2$, where e_s, e_r and e_o are the real-valued embedding vectors for the subject, relationship and object of a fact. Further details are given by Bordes et al. (2013).

Embeddings for entities with few connections are easy to learn, yet the quality of these embeddings depends on how inter-connected they are. In the extreme case where the subject and object of a triple only appears once in the dataset, the learned embeddings of the subject and object will be semantically meaningless. This happens very often in SimpleQuestions, since only around 5% of the entities have more than 2 connections in the graph. Thus, by applying TransE directly over this set of triples, we would eventually end up with a layout of entities that does not contain clusters of semantically close concepts. In order to guarantee an effective semantic representation of the embeddings, we have to learn them together with additional triples extracted from the whole Freebase graph to complement the SimpleQuestions graph with relevant information for this task.

We need a coarse representation for the entities contained in SimpleQuestions, capturing the *basic* information, like the profession or nationality, the annotators tend to use when phrasing the questions, and accordingly we have ensured the embeddings contain this information by taking triples coming from the Freebase graph[5] regarding:

1. Category information: given by the *type/instance* relationship, this ensures that all the entities of the same semantic category are close to each other. Although one might think that the expected category of the subject/object could be inferred directly from the relationship, there are fine-grained differences in the expected types that be extracted only directly by observing this category information.

2. Geographical information: sometimes the annotators have included information about nationality (e.g. *Which French president... ?*) or location (e.g. *Where in Germany...* ?) of the subject and/or object. This information is given by the relationships *person/nationality* and *location/contained_by*. By including these facts in the learning, we ensure the existence of a fine-grained layout of the embeddings regarding this information within a same category.

[5] Extracted from one of the latest Freebase dumps (downloaded in mid-August 2015) `https://developers.google.com/freebase/data`

Closest neighbors to	Warner Bros. Entertainment	Manchester	hindi language
SQ	Billy Gibbons Jenny Lewis Lies of Love Swordfish	Ricky Anane Lee Dixon Jerri Bryne Greg Wood	nepali indian Naseeb Ghar Ek Mandir standard chinese
SQ + FB	Paramount Pictures Sony Pictures Entertainment Electronic Arts CBS	Oxford Sale Liverpool Guildford	dutch language italian language danish language bengali language

Table 2: Examples of differences in the local structure of the vector space embeddings when adding more FB facts

3. Gender: similarly, sometimes annotators have included information about gender (e.g. *Which male audio engineer...?*). This information is given by the relationship *person/gender*.

To this end, we have included more than $300,000$ facts from Freebase in addition to the facts in SimpleQuestions for training. Table 2 shows the differences in the embeddings before and after adding additional facts for training the TransE representations.

4.4 Generating Questions

To resolve the problem of data sparsity and previously unseen words on the target side, we draw inspiration from the placeholders proposed for handling rare words in neural machine translation by Luong et al. (2015). For every question and answer pair, we search for words in the question which overlap with words in the subject string of the fact.[6] We heuristically estimate the sequence of most likely words in the question, which correspond to the subject string. These words are then replaced by the placeholder token *<placeholder>*. For example, given the fact {fires_creek, contained_by, nantahala_national_forest} the original question *Which forest is Fires Creek in?* is transformed into the question *Which forest is <placeholder>in?*. The model is trained on these modified questions, which means that model only has to learn decoder embeddings for tokens which are not in the subject string. At test time, after outputting a question, all placeholder tokens are replaced by the subject string and then the outputs are evaluated. We call this the Single-Placeholder (SP) model. The main difference with respect to that of Luong et al. (2015) is that we do not use placeholder tokens in the input language, because then the entities and relationships in the input would not be able to transmit semantic (e.g. topical) information to the decoder. If we had included placeholder tokens in the input language, the model would not be able to generate informative words regarding the subject in the question (e.g. it would be impossible for the model to learn that the subject *Paris* may be accompanied by the words *French city* when generating a question, because it would not see *Paris* but only a placeholder token).

A single placeholder token for all question types could unnecessarily limit the model. We therefore also experiment with another model, called the Multi-Placeholder (MP) model, which uses 60 different placeholder tokens such that the placeholder for a given question is chosen based on the subject category extracted from the relationship (e.g. contained_by is classified in the category location, and so the transformed question would be *Which forest is <location placeholder> in?*). This could make it easier for the model to learn to phrase questions about a diverse set of entities, but it also introduces additional parameters, since there are now 60 placeholder embeddings to be learned, and therefore the model may suffer from overfitting. This way of addressing the sparsity in the output reduces the vocabulary size to less than 7000 words.

4.5 Template-based Baseline

To compare our neural network models, we propose a (non-parametric) template-based baseline model, which makes use of the entire training set when generating a question. The baseline operates on questions modified with the placeholder as in the preceding section. Given a fact F as input, the baseline picks a candidate fact F_c in the training set at uniformly random, where F_c has the same relationship as F. Then the baseline considers the questions corresponding to F_c and as in the

[6] We use the tool difflib: `https://docs.python.org/2/library/difflib.html`.

SP model, in the final step the placeholder token in the question is replaced by the subject string of the fact F.

5 Experiments

5.1 Training Procedure

All neural network models were implemented in Theano (Theano Development Team, 2016). To train the neural network models, we optimized the log-likelihood using the first-order gradient-based optimization algorithm Adam (Kingma and Ba, 2015). To decide when to stop training we used early stopping with patience (Bengio, 2012) on the METEOR score obtained for the validation set. In all experiments, we use the default split of the SimpleQuestions dataset into training, validation and test sets.

We trained TransE embeddings with embedding dimensionality 200 for each subject, relationship and object. Based on preliminary experiments, for all neural network models we fixed the learning rate to 0.00025 and clipped parameter gradients with norms larger than 0.1. We further fixed the embedding dimensionality of words to be 200, and the hidden state of the decoder RNN to have dimensionality 600.

5.2 Evaluation

To investigate the performance of our models, we make use of both automatic evaluation metrics and human evaluators.

5.2.1 Automatic Evaluation Metrics

BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) are two widely used evaluation metrics in statistical machine translation and automatic image-caption generation (Chen et al., 2015). Similar to statistical machine translation, where a phrase in the source language is mapped to a phrase in the target language, in this task a KB fact is mapped to a natural language question. Both tasks are highly constrained, e.g. the set of valid outputs is limited. This is true in particular for short phrases, such as one sentence questions. Furthermore, in both tasks, the majority of valid outputs are paraphrases of each other, which BLEU and METEOR have been designed to capture. We therefore believe that BLEU and METEOR constitute reasonable performance metrics for evaluating the generated questions.

Although we believe that METEOR and BLEU are reasonable evaluation metrics, they may have not recognize certain paraphrases, in particular paraphrases of entities. We therefore also make use of a sentence similarity metric, as proposed by Rus and Lintean (2012), which we will denote *Embedding Greedy* (Emb. Greedy). The metric makes use of a word similarity score, which in our experiments is the cosine similarity between two Word2Vec word embeddings (Mikolov et al., 2013).[7] The metric finds a (non-exclusive) alignment between words in the two questions, which maximizes the similarity between aligned words, and computes the sentence similarity as the mean over the word similarities between aligned words.

The results are shown in Table 3. Example questions produced by the model with multiple placeholders are shown in Table 4. The neural network models outperform the template-based baseline by a clear margin across all metrics. The template-based baseline is already a relatively strong model, because it makes use of a separate template for each relationship. Qualitatively the neural networks outperform the baseline model in cases where they are able to levage additional knowledge about the entities (see first, third and fifth example in Table 4). On the other hand, for rare relationships the baseline model appears to perform better, because it is able to produce a reasonable question if only a single example with the same relationship exists in the training set (see eighth example in Table 4). Given enough training data this suggests that neural networks are generally better at the question generation task compared to hand-crafted template-based procedures, and therefore that they may be useful for generating question answering corpora. Furthermore, it appears that the best performing models are the models where TransE are trained on the largest set of triples (TransE++). This set contains, apart from the supporting triples described in Section 4.3, triples involving entities which are highly connected to the entities found in the SimpleQuestions facts. In total, around 30 millions of facts, which have been used to generate the 30M Factoid Question-Answer Corpus. Lastly, it is not clear whether the model with a single placeholder or the model with multiple placeholders performs best. This motivates the following human study.

[7] We use the Word2Vec embeddings pretrained on the Google News Corpus: `https://code.google.com/p/word2vec/`.

Model	BLEU	METEOR	Emb. Greedy
Baseline	31.36	33.12	74.02
SP Triples	33.27	35.07	76.72
MP Triples	32.76	34.97	76.70
SP Triples TransE++	**33.32**	**35.38**	76.78
MP Triples TransE++	33.28	35.29	**77.01**

Table 3: Test performance for all models w.r.t. BLEU, METEOR and Emb. Greedy performance metrics, where *SP* indicates models with a single placeholder and *MP* models with multiple placeholders. *TransE++* indicates models where the TransE embeddings have been pretrained on a larger set of triples. The best performance on each metric is marked in bold font.

Fact	Human	Baseline	MP Triples TransE++
bayuvi dupki – contained by – europe	where is bayuvi dupki?	what state is the city of bayuvi dupki located in?	what continent is bayuvi dupki in?
illinois – contains – ludlow township	what is in illinois?	what is a tributary found in illinois?	what is the name of a place within illinois?
neo contra – publisher – konami	who published neo contra?	which company published the game neo contra?	who is the publisher for the computer videogame neo contra?
fumihiko maki – structures designed – makuhari messe	fumihiko maki designed what structure?	what park did fumihiko maki help design?	what's a structure designed by fumihiko maki?
cheryl hickey – profession – actor	what is cheryl hickey's profession?	what is cheryl hickey?	what is cheryl hickey's profession in the entertainment industry?
cherry – drugs with this flavor – tussin expectorant for adults 100 syrup	name a cherry flavored drug?	what is a cherry flavored drug?	what's a drug that cherry shaped like?
pop music – artists – nikki flores	what artist is known for pop music?	An example of pop music is what artist?	who's an american singer that plays pop music?

Table 4: Test examples and corresponding questions.

5.2.2 Human Evaluation Study

We carry out pairwise preference experiments on Amazon Mechanical Turk.

Initially, we considered carrying out separate experiments for measuring relevancy and fluency respectively, since this is common practice in machine translation. However, the relevancy of a question is determined solely by a single factor, i.e. the relationship, since by construction the subject is always in the question. Measuring relevancy is therefore not very useful in our task. To verify this we carried out an internal pairwise preference experiment with human subjects, who were repeatedly shown a fact and two questions and asked to select the most relevant question. We found that 93% of the questions generated by the MP Triples TransE++ model were either judged better or at least as good as the human generated questions w.r.t. relevancy. The remaining 7% questions of the MP Triples TransE++ model questions were also judged relevant questions, although less so compared to the human generated questions. In the next experiment, we therefore measure the holistic quality of the questions.

We setup experiments comparing: Human-Baseline (human and baseline questions), Human-MP (human and MP Triples TransE++ questions) and Baseline-MP (baseline and MP Triples TransE++ questions). We show human evaluators a fact along with two questions, one question from each model for the corresponding fact, and ask the them to choose the question which is most relevant to the fact and most natural. The human evaluator also has the option of not choosing either question. This is important if both questions are equally good or if neither of the questions make sense. At the beginning of each experiment, we show the human evaluators two examples of statements and a corresponding pair of questions, where we briefly explain the form of the statements and how questions relate to those statements. Following the introductory examples, we present the facts and cor-

Model A	Model B	Model A Preference (%)	Model B Preference (%)	Fleiss' kappa
Human	Baseline	$^{*}\mathbf{56.329 \pm 5.469}$	34.177 ± 5.230	0.242
Baseline	MP Triples TransE++	32.484 ± 5.180	$^{*}\mathbf{60.828 \pm 5.399}$	0.234
Human	MP Triples TransE++	38.652 ± 5.684	$\mathbf{51.418 \pm 5.833}$	0.182

Table 5: Pairwise human evaluation preferences computed across evaluators with 95% confidence intervals. The preferred model in each experiment is marked in bold font. An asterisk next to the preferred model indicates a statistically significance likelihood-ratio test, which shows that the model is preferred in at least half of the presented examples with 95% confidence. The name *MP Triples TransE++* indicates the model with multiple placeholders and TransE embeddings pretrained on a larger set of triples. The last column shows the Fleiss' kappa averaged across batches (HITs) with different evaluators and questions.

responding pair of questions one by one. To avoid presentation bias, we randomly shuffle the order of the examples and the order in which questions are shown by each model. During each experiment, we also show four check facts and corresponding check questions at random, which any attentive human annotator should be able to answer easily. We discard responses of human evaluators who fail any of these four checks.

The preference of each example is defined as the question which is preferred by the majority of the evaluators. Examples where neither of the two questions are preferred by the majority of the evaluators, i.e. when there is an equal number of evaluators who prefer each question, are assigned to a separate preference class called "comparable".[8]

The results are shown in Table 5. In total, $3,810$ preferences were recorded by 63 independent human evaluators. The questions produced by each model model pair were evaluated in 5 batches (HITs). Each human evaluated 44-75 examples (facts and corresponding question pairs) in each batch and each example was evaluated by 3-5 evaluators. In agreement with the automatic evaluation metrics, the human evaluators strongly prefer either the human or the neural network model over the template-based baseline. Furthermore, it appears that humans cannot distinguish between the human-generated questions and the neural network questions, on average showing a preference towards the later over the former ones. We hypothesize this is because our model penalizes uncommon and unnatural ways to frame questionsand sometimes, includes specific information about the target object that the humans do not (see last example in Table 4). This confirms our earlier assertion, that the neural network questions can be used for building question answering systems.

6 Conclusion

We propose new neural network models for mapping knowledge base facts into corresponding natural language questions. The neural networks combine ideas from recent neural network architectures for statistical machine translation, as well as multi-relational knowledge base embeddings for overcoming sparsity issues and placeholder techniques for handling rare words. The produced question and answer pairs are evaluated using automatic evaluation metrics, including BLEU, METEOR and sentence similarity, and are found to outperform a template-based baseline model. When evaluated by untrained human subjects, the question and answer pairs produced by our best performing neural network appears to be comparable in quality to real human-generated questions. Finally, we use our best performing neural network model to generate a corpus of 30M question and answer pairs, which we hope will enable future researchers to improve their question answering systems.

Acknowledgments

The authors acknowledge IBM Research, NSERC, Canada Research Chairs and CIFAR for funding. The authors thank Yang Yu, Bing Xiang, Bowen Zhou and Gerald Tesauro for constructive feedback, and Antoine Bordes, Nicolas Usunier, Sumit Chopra and Jason Weston for providing the SimpleQuestions dataset. This research was enabled in part by support provided by Calcul Qubec (`www.calculquebec.ca`) and Compute Canada (`www.computecanada.ca`).

[8] The probabilities for the "comparable" class in Table 5 can be computed in each row as 100 minus the third and fourth column in the table.

References

[Ali et al.2010] Husam Ali, Yllias Chali, and Sadid A Hasan. 2010. Automation of question generation from sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 58–67.

[Bahdanau et al.2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

[Banerjee and Lavie2005] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL, Workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

[Bengio2012] Yoshua Bengio. 2012. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer.

[Berant and Liang2014] Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *ACL*, pages 1415–1425.

[Bollacker et al.2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

[Bordes et al.2011] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. 2011. Learning structured embeddings of knowledge bases. In *AAAI 2011*.

[Bordes et al.2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

[Bordes et al.2014] Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Machine Learning and Knowledge Discovery in Databases - European Conference, (ECML PKDD)*, pages 165–180.

[Bordes et al.2015] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

[Chen et al.2009] Wei Chen, Gregory Aist, and Jack Mostow. 2009. Generating questions automatically from informational text. In *Proceedings of the 2nd Workshop on Question Generation (AIED 2009)*, pages 17–24.

[Chen et al.2015] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

[Cho et al.2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

[Curto et al.2012] Sergio Curto, A Mendes, and Luisa Coheur. 2012. Question generation based on lexico-syntactic patterns learned from the web. *Dialogue and Discourse*, 3(2):147–175.

[Duma and Klein2013] Daniel Duma and Ewan Klein. 2013. Generating natural language from linked data: Unsupervised template extraction. *ACL*, pages 83–94.

[Dumais et al.2002] Susan Dumais, Michele Banko, Eric Brill, Jimmy Lin, and Andrew Ng. 2002. Web question answering: Is more always better? In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298.

[Graesser et al.1992] Arthur C Graesser, Sallie E Gordon, and Lawrence E Brainerd. 1992. QUEST: A model of question answering. *Computers and Mathematics with Applications*, 23(6):733–745.

[Greff et al.2015] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2015. LSTM: A search space odyssey. *arXiv preprint arXiv:1503.04069*.

[Kalady et al.2010] Saidalavi Kalady, Ajeesh Elikkottil, and Rajarshi Das. 2010. Natural language question generation using syntax and keywords. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 1–10. questiongeneration. org.

[Kingma and Ba2015] Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

[Lenat and Guha1989] Douglas B. Lenat and Ramanathan V. Guha. 1989. *Building large knowledge-based systems; representation and inference in the Cyc project*. Addison-Wesley Longman Publishing Co., Inc.

[Lopez et al.2011] Vanessa Lopez, Victoria Uren, Marta Sabou, and Enrico Motta. 2011. Is question answering fit for the semantic web? a survey. *Semantic Web*, 2(2):125–155.

[Luong et al.2015] Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *ACL*, pages 11–19.

[Mannem et al.2010] Prashanth Mannem, Rashmi Prasad, and Aravind Joshi. 2010. Question generation from paragraphs at upenn: Qgstec system description. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 84–91.

[Mikolov et al.2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

[Olney et al.2012] Andrew M Olney, Arthur C Graesser, and Natalie K Person. 2012. Question generation from concept maps. *Dialogue and Discourse*, 3(2):75–99.

[Papineni et al.2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

[Rus and Lintean2012] Vasile Rus and Mihai Lintean. 2012. A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, NAACL*, pages 157–162.

[Rus et al.2010] Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The first question generation shared task evaluation challenge. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 251–257.

[Sutskever et al.2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*.

[Theano Development Team2016] Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May.

[Voorhees and Tice2000] Ellen M Voorhees and DM Tice. 2000. Overview of the trec-9 question answering track. In *TREC*.

[Vrandečić and Krötzsch2014] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

[Yao and Zhang2010] Xuchen Yao and Yi Zhang. 2010. Question generation with minimal recursion semantics. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 68–75.

[Yao et al.2012] Xuchen Yao, Gosse Bouma, and Yi Zhang. 2012. Semantics-based question generation and implementation. *Dialogue and Discourse*, 3(2):11–42.