Generating Questions Based on Semi-Automated and End-to-End Neural Network

Tianci Xia[1, 2], Yuan Sun[1, 2, *], Xiaobing Zhao[1, 2], Wei Song[1] and Yumiao Guo[3]

Abstract: With the emergence of large-scale knowledge base, how to use triple information to generate natural questions is a key technology in question answering systems. The traditional way of generating questions require a lot of manual intervention and produce lots of noise. To solve these problems, we propose a joint model based on semi-automated model and End-to-End neural network to automatically generate questions. The semi-automated model can generate question templates and real questions combining the knowledge base and center graph. The End-to-End neural network directly sends the knowledge base and real questions to BiLSTM network. Meanwhile, the attention mechanism is utilized in the decoding layer, which makes the triples and generated questions more relevant. Finally, the experimental results on SimpleQuestions demonstrate the effectiveness of the proposed approach.

Keywords: Generating questions, semi-automated model, End-to-End neural network, question answering.

1 Introduction

Question answering (QA) over knowledge base (KB) has attracted more and more researchers' attention in recent years. One of the main reasons is the rapid development of large-scale knowledge base, such as Freebase [Bollacker, Cook and Tufts (2007)], DBpedia [Hellmann, Morsey, Van Kleef et al. (2015)], and so on. However, QA systems need large-scale question-answer pairs, so how to generate questions based on KB has become a difficult problem in the field of natural language processing (NLP). At present, there are two main ways to solve this problem. One is template-based method [Su, Sun, Sadler et al. (2016); Chen, Aist and Mostow (2009); Ali, Chali and Hasan (2010)], and the other is End-to-End neural network methods [Serban, GarcÃadurÃ¡n, Gulcehre et al. (2016)].

The template-based method needs to build question templates manually and replace the entity position with special placeholder. Then using entities in KB to replace the

[1] The School of Information Engineering, Minzu University of China, Beijing, 100081, China.

[2] Minority Languages Branch, National Language Resource and Monitoring Research Center, Beijing, 100081, China.

[3] Queen's University Belfast, Belfast, Northern Ireland, BT71NN, UK.

[*] Corresponding Author: Yuan Sun. Email: tracy.yuan.sun@gmail.com.

CMC. doi:10.32604/cmc.2019.05860 www.techscience.com/cmc

placeholder of the original position, and generating the question after manual verification. These questions meet the requirement of the QA system, and there are almost no grammatical and semantic errors. But the process of constructing templates is time consuming, and the generalization of templates is very poor. The End-to-End neural network method directly sends triples of KB and questions to the neural network for training. After a certain of iterations, questions corresponding to triples are generated. This method is simple and does not require manual intervention. But the generated questions often have some grammatical and semantic errors. In our work, we avoid shortcomings of two ways. Firstly, we use the semi-automated model based on KB to generate question templates and real questions automatically. Then, we send the KB and real questions to the Bi-directional long short-term memory (BiLSTM) neural network to generate questions related to triples. Finally, we compare the questions generated by two models and choose the one with larger bilingual evaluation understudy (BLEU) [Papineni, Roukos, Ward et al. (2002)] value as the final output.

The contributions of this paper can be summarized as follows:

(1) We use center graph to automatically build question templates. Using templates and KB to make the question generation smoother.

(2) We use KB and End-to-End neural network to automatically generate questions. Also, the attention mechanism is utilized in the decoding layer to make triples and generated questions more relevant.

(3) We choose better questions by comparison automatically.

2 Related work

In recent years, the field of generating questions has attracted more and more attention. Some template-based methods have been proposed. Ali proposed *wh-inversion* and *wh-fronting* methods which use the way of matching content entities like “who”, “where” to generate questions directly [Ali, Chali and Hasan (2010)]. However, it does not make full use of the semantic information in the content. The semantic information can be obtained through named entity recognition [Mannem, Prasad and Joshi (2010); Yao (2010)] and semantic role labeling [Chen, Aist and Mostow (2009)]. In Mendes et al. [Mendes, Piwek, and Boyer (2012)], the question is classified according to its syntactic structure, answer type and question prefix. Then the model learns patterns to generate questions. After identifying keywords in the question, Graesser et al. [Graesser, Gordon and Brainerd (1992)] classified questions according to 18 predefined categories. On this basis, Chen et al. [Chen, Aist and Mostow (2009)] generated target questions by handcrafted-templates. Recently, the way of building templates integrating KB has become a research hotspot. There are some KB-based QA datasets constructed by manual annotation, such as QALD [Lopez, Unger, Cimiano et al. (2013)], FREE917 [Cai and Yates (2013)], SimpleQuestions [Petrochuk and Zettlemoyer (2018)], and so on.

Another effective way to construct QA dataset is based on End-to-End neural network model [Serban, GarcÃadurÃ, Gulcehre et al. (2016)], which uses the recurrent neural network to automatically formulate questions based on KB. Olney et al. [Olney, Graesser and Person (2012)] took triples as inputs to Long Short-Term Memory (LSTM) network.

Subject entities of triples are replaced by placeholders in pre-defined position. Similarly, Duma et al. [Duma and Klein (2013)] fed the triple directly into the neural network and generated short text descriptions with placeholders, which can generate fact questions by filling slots. In addition, with the development of deep learning methods, Generative Adversarial Networks (GANs) are gradually used in NLP tasks, such as text generation [Yu, Zhang, Wang et al. (2017); Li, Monroe, Shi et al. (2017)], distant supervised relation extraction [Zeng, Dai, Li et al. (2018)], and so on. These works can be used for references in question generation.

Similar to Olney, our baseline model mainly defines some relational templates and uses placeholders to replace head entities. The main differences are that our model optimizes question templates by center graph, and uses the question selector to choose the best question as output. At the same time, in order to optimize the neural network model, we feed back the output to End-to-End model and get better parameters.

3 Model architecture

The overall framework of the joint model is shown in Fig. 1. It is mainly divided into four parts: KB preprocessing, semi-automated model, End-to-End neural network and question selector.

The KB preprocessing is divided into two subtasks: center entity set for semi-automated model and triple embedding for End-to-End neural network.

The semi-automated model combines the center graph and ontology to generate question templates and real questions. The End-to-End neural network mainly encodes the triple of KB to BiLSTM, and uses the attention mechanism in the decoding layer to generate the related questions. Finally, we choose the question with larger BLEU value as the final output by the question selector.

3.1 Preprocessing for knowledge base

In this section, we treat the KB as two subtasks respectively: center entity set for semi-automated model and triple embedding for End-to-End neural network. Firstly, we define the triple in KB as <Subject, Relation, Object>. It can be abbreviated as $<s,r,o>$. The semi-automated model can find all triples in KB associated with the center entity like Yao Ming in Fig.1. It should be noted that if the number of selected triples is too small, we can find other entities of the same type as the center entity. For example, according the center entity Yao Ming, we get three triples related to it: <Yao Ming, wife, Ye Li>, <Yao Ming, birthtime, 1980>, and <Yao Ming, birthplace, ShangHai>. Then we retrieve all entities in the three triples and find the entity Ye Li has the same type as center entity, so we traverse with Ye Li and get a new triple <Ye Li, birthplace, ShangHai>. We use the TransE [Bordes, Usunier, García-Durán et al. (2013)] algorithm to map the triple <Yao Ming, wife, Ye Li> into the low-dimensional dense vector space, and get the triple embedding $<Enc$(Yao Ming), Enc(wife), Enc(Ye Li)>. Enc(Yao Ming), Enc(wife), Enc(Ye Li) are embeddings of subject entity Yao Ming, relation wife and object entity Ye Li. And put the triple embedding $<Enc$(Yao Ming), Enc(wife), Enc(Ye Li)> to End-to-End neural network.

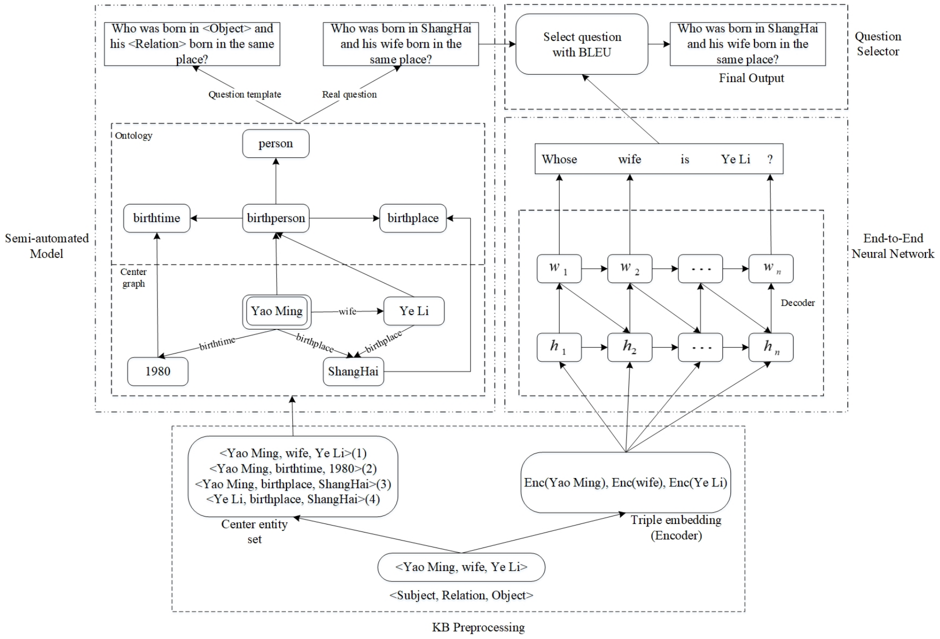
Who was born in <Object> and his <Relation> born in the same place?
Who was born in ShangHai and his wife born in the same place?
Question template
Real question
Select question with BLEU
Who was born in ShangHai and his wife born in the same place?
Final Output
Question Selector
Ontology
person
birthtime
birthperson
birthplace
Semi-automated Model
Center graph
Yao Ming
wife
Ye Li
birthtime
birthplace
birthplace
1980
ShangHai
Whose
wife
is
Ye Li
?
End-to-End Neural Network
w_1
w_2
...
w_n
Decoder
h_1
h_2
...
h_n
<Yao Ming, wife, Ye Li>(1)
<Yao Ming, birthtime, 1980>(2)
<Yao Ming, birthplace, ShangHai>(3)
<Ye Li, birthplace, ShangHai>(4)
Center entity set
Enc(Yao Ming), Enc(wife), Enc(Ye Li)
Triple embedding (Encoder)
<Yao Ming, wife, Ye Li>
<Subject, Relation, Object>
KB Preprocessing

Figure 1: The overall framework of the joint model

3.2 Semi-automated model

3.2.1 Center graph

Inspired by the structure diagram of KB, the center graph method is adopted. It can visualize the syntax information and the semantic information. For the syntax information, the graph query q is built on a given KB K . It consists of three nodes: center node (double rounded rectangle), indirect node (rounded rectangle) and direct node (shaded rounded rectangle). For the semantic information, graph query is the proper subset of λ -calculus shown in the following.

$$\lambda x.\exists y.\exists z.type(x, birthperson)$$

$$\wedge type(y, birthperson)$$

$$\wedge type(z, birthtime) \wedge wife(x, y)$$

$$\wedge birthplace(x, ShangHai)$$

$$\wedge birthplace(y, ShangHai)$$

$$\wedge birthtime(x, z) \wedge (z, 1980)$$

The answer of the question can be easily obtained from the center node. For example, the center node "Yao Ming" is the answer to the question "Who was born in ShangHai and his wife born in the same place?", shown in Fig. 1. Different from [Yih, Chang, He et al. (2015)], graph query has good expression which is not restricted by tree structure.

3.2.2 Generating question templates and real questions

This section mainly describes how to generate question templates and real questions. The semi-automated model takes triples and ontology as input. Also, the operations are random to prevent the biases in generating questions. For example, we first select the birthperson node as the center node. Then traversing other nodes and edges connected to the center node in turn, we get question templates by combining these nodes. In addition, according to the ontology, the node is replaced with the corresponding entity to get the real question.

Functional words are very important components in the process of generating questions. Functional words generally refer to the interrogative words corresponding to the center entity. In Fig.1, the functional words corresponding to Yao Ming may be "who" or "whose". These functional words are predefined and frequently used, such as "who", "what", "whose", "when", "where" and so on.

When the question template is generated, we use the question template to directly match the triples of the same relation in the KB to get more questions. And the answer to the question is the center entity of the current triple, described in Section 3.2.1.

3.3 End-to-End neural network model

We address this task with a model inspired by the recent successful neural machine translation model [Sutskever, Vinyals and Le (2014); Neco and Forcada (1997)]. We can regard the task as "differential translation", that is, the transformation from KB to natural language question, while ignoring the potential structured knowledge in the model. These models usually consist of two parts: encoding layer which mainly encodes the source triples into a fixed size intermediate vector and decoding layer which decodes the intermediate vector into the target question.

3.3.1 Encoding layer

Contrary to the neural machine translation network, our source language is not a natural language, but triples in KB. We use the TransE algorithm, which maps each element of the triple into embedding. Each element in fact $\{s,r,o\}$ represents the subject entity, the relation and the object entity respectively. At the same time, the fact $F=(s,r,o)$ is mapped to a embedding of 1-of-k vector, which is obtained by $e_f = Ein_{(\mathrm{s,r,o})}$, where Ein is the embedding matrix of the whole input entities, and k is the vector size of the vocabulary.

The embedding matrix Ein is a parameter to learn from the model. We have learned it in advance through the TransE algorithm. We adjust it in the pre-training process, and do not allow the model to be modified during the model training process.

3.3.2 Decoding layer

In the decoding layer, we use the BiLSTM with an attention-mechanism [Bahdanau, Cho and Bengio (2014)] to decode the output of the encoding layer into question Q related to the fact F. Recently, BiLSTM is more effective and has achieved good results in many fields, which compared with other recurrent neural network architectures, such as LSTM

[Greff, Srivastava, Koutník et al. (2017)] and Gated Recurrent Unit (GRU) [Chung, Ãaglar GülÃ§ehre, Cho et al. (2014)]. The time step *n* of the decoding layer in each hidden state of BiLSTM is computed (forward is the same as backward), as shown in Eq. (1)-Eq. (4).

$$i_t = \sigma(W_i E_{out} + C_i c(F, h_{n-1}) + U_i h_{n-1}) \tag{1}$$

$$f_t = \sigma(W_f E_{out} + C_f c(F, h_{n-1} + U_f h_{n-1})) \tag{2}$$

$$h_d = \tanh(W E_{out} w_{n-1} + C c(F, h_{n-1}) + U(i_t o h_{n-1})) \tag{3}$$

$$h_n = f_t o h_{n-1} + (1 - f_t) o h_d \tag{4}$$

where i_t, f_t represent input gate and forget gate respectively. σ is the sigmoid function and o is multiplication of corresponding elements. $E_{out} W_n$ is the decoding embedding vector corresponding to the word W_n, the variables $U_i, U_f, C_i, C, W_i, W_f, W$ are the training parameters of the BiLSTM and $c(F, h_{n-1})$ is the context mapping vector. The initial state h_0 is given by the output of a forward neural network which is fed into the fact embedding. And the final output h while concatenates the forward $\overrightarrow{h_n}$ and the backward $\overleftarrow{h_n}$. At each time step n, the probability distribution corresponding to each word is output through softmax layer. At last, the function $c(F, h_{n-1})$ is computed by adding the attention mechanism, shown in Eq. (5).

$$c(F, h_{n-1}) = a_{s,n-1} Enc(F)_s + a_{r,n-1} Enc(F)_r + a_{o,n-1} Enc(F)_o \tag{5}$$

where $a_{s,n-1}, a_{r,n-1}, a_{o,n-1}$ are mainly the real-value scalars, used to measure the weight of the corresponding subject entity, relation and object entity. $Enc(F)_s$, $Enc(F)_r$ and $Enc(F)_o$ are embedding vectors of corresponding elements $< s, r, o >$.

3.3.3 Generating questions

In order to solve the problem of sparse triples in KB, we draw on the idea of using placeholders in neural machine translation by Luong et al. [Luong, Sutskever, Le et al. (2014)]. For each question and answer pair, we try to match the words in the question with the subject entity in the triple as much as possible. We estimate the sequence of words most likely to occur in the question from a heuristic perspective, and they correspond to the subject entity.

At the same time, we use the questions selected in question selector to optimize the parameters of the End-to-End neural network model. We use the selected questions as the standard answers in the training set, and the generated questions as the predictive answers. By optimizing the parameters, we can narrow the difference between the two results.

3.4 Question selector

We compare questions generated from the semi-automated model and End-to-End neural network. Using the BLEU value to choose better questions. The question selector has two functions. On the one hand, the selected question can be used as the final output of the whole model. On the other hand, the selected question can be sent into the neural network for parameter optimization. Also, human evaluation is important to help optimize the question structure and paraphrasing. Finally, we construct the QA pairs.

4 Experiments

4.1 Dataset

We use the SimpleQuestions dataset [Petrochuk and Zettlemoyer (2018)] to train our model. SimpleQuestions is mainly based on Freebase triples, using manual annotation to map the triples and questions. It contains over 100K Freebase English QA pairs. Topics and relationships are clearly presented in each question.

4.2 Training

All neural network models were implemented in Tensorflow [Abadi (2016)]. In order to train our model effectively, we use gradient optimization algorithm Adam [Kingma and Ba (2015)] for maximum likelihood estimation. At the same time, in order to accelerate convergence, we use Practical Recommendations method [Bengio (2012)] to adjust parameters in the verification set. In this paper, the Simple Questions dataset is divided into training set, validation set, and testing set. We train the TransE algorithm for each subject entity, relation and object entity. We set the learning rate to 0.01 initially, which makes the model unable to converge. So, we try to reduce this value. When the value is 0.00001, the model can converge. We further set the embedding dimension of the word to 200 and the hidden state of the decoder to 600. And we initially set the num of units to 32, but the training speed is too slow. Therefore, we reduce it to 16 under the premise of guaranteeing the result. Hyper parameters for network training are shown in Tab. 1.

Table 1: Hyper parameters setting

Parameter name	Parameter value
hidden state numbers	600
learning rate	0.00001
keep prob	1.0
batch size	64
patience	20
embedding dim	200
num units	16

4.3 Evaluation

In order to verify the validity of our model, we evaluate it using two methods: automated evaluation and human evaluation.

4.3.1 Automatic evaluation

BLEU and METEOR [Bahdanau, Cho and Bengio (2014)] are two commonly used evaluation indicators in machine translation and image-caption generation [Satanjeev (2005)]. Similar to machine translation, the main evaluation indicators in this paper are the mapping of BLEU to natural questions from KB. It is especially effective for the short sentence. Furthermore, most of the effective outputs in both missions are mutually interpreted, BLEU and METEOR are mainly used to evaluate the correlation between input and output. Thus, BLEU and METEOR can be used to evaluate the correlation between triples and questions.

In this paper, we use BLEU as the evaluation because it performs well on statements and matches well at the corpus level. Firstly, we introduce the Brevity Penalty (BP) value which is computed in Eq. (6).

$$BP = \begin{cases} 1, & \text{if } c > r \\ e^{1-r/c}, & \text{if } c <= r \end{cases} \tag{6}$$

where c is the length of the generating question, r is the length of the real question. And the BLEU value is computed shown in Eq. (7).

$$BELU = BP * \exp(\sum_{n=1}^{N} w_n \log p_n) \tag{7}$$

where p_n is the n-gram value, w_n is the context sliding window. We set the w_n is 1/4, that is, this evaluation indicator is BLEU-4.

The BLEU-4 value in each epoch is shown in Fig. 2. At the 10th epoch, the BiLSTM + Attention model is clearly better than others. After the 10th epoch, each method is obviously improved, and the Semi-automated + BiLSTM + Attention model (our model) is better than others. By comparing the generated questions of BiLSTM with attention model and Semi-automated model, we select the questions with higher BLEU-4 value as the final output. The experimental results of different models are shown in Tab. 2. Compared with Semi-automated, LSTM, BiLSTM and BiLSTM + Attention, our model increases BLEU-4 value by 84.2%, 56.9%, 32.6% and 4.9% respectively. And through Tab. 3, we can see that the generated questions are no longer the simple questions, which have higher quality of our model than others in human evaluation.

Table 2: The BLEU-4 value in various models

Model	BLEU-4
Semi-automated [Su, Sun, Sadler et al. (2016)]	9.28
LSTM	10.9
BiLSTM	12.9
BiLSTM+Attention	16.3
BiLSTM+Attention+Semi-automated	17.1

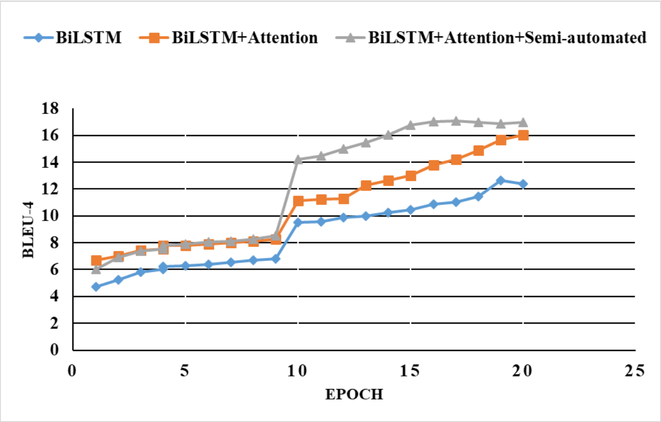

Figure 2: Three models for each epoch

4.3.2 Human evaluation

Also, we consider the results for relevance and fluency separately. The most important part of the question depends on the relation in the triple, because the generating questions should be related to the relation and contain the semantic information of the relation. In order to validate this point, we try to evaluate the relevance of the question and relation to verify whether the question is good or bad. Tab. 3 shows the results which using a fact to generate the most relevant question by different models.

Table 3: Examples and corresponding questions

Fact	Human	Semi-automated	BiLSTM+ Attention	BiLSTM+ Attention+ Semi-automated
<pop music, artistsnikki, flores>	What artist is known for pop music?	Which artist is pop music?	Who is a singer that plays pop music?	Who is a singer that plays pop music?
<Yao Ming, wife, Ye Li>	Whose wife is the basketball player Ye Li?	Who is born in Shanghai and his wife born in the same place?	Whose wife is Ye Li?	Who is born in Shanghai and his wife born in the same place?
<Terrorism, attacks, September 11>	The September were carried out with the involvement of what terrorist organizations?	Who did September 11 attacks?	Find who organizations involved in September 11 attacks?	Find who organizations involved in September 11 attacks?
<Yao Beina, death cause,	Who died of Cancer?	What reason did Yao Beina die	Who died of Cancer and	Who died of Cancer and was

Cancer>		of?	was a singer?	a singer?
<Venice, location, Italy>	Name a location in Italy?	Where is the Venice and is a beautiful city?	Which city is beautiful in Italy?	Where is the Venice and is a beautiful city?
<Prince adventure, written work, subjects>	What subjects is racing to the beginning of the road written about?	What are the subjects?	What is about beginning of the road written?	What is about beginning of the road written?

5 Conclusion

We propose a joint model based on semi-automated and End-to-End neural network for mapping KB facts into corresponding natural language questions. The End-to-End neural network with the attention mechanism is utilized to generate questions, which make the triples and generated questions more relevant. Generated questions are evaluated by BLEU-4 value and human evaluation. The experimental results show the effectiveness of the proposed method.

Acknowledgement: This work is supported by National Nature Science Foundation (No. 61501529, No. 61331013), National Language Committee Project of China (No. ZDI125-36) and Young Teachers' Scientific Research Project in Minzu University of China.

References

Abadi, M. (2016): Tensorflow: learning functions at scale. *ACM Sigplan Notices*, vol. 51, no. 9, pp. 1.

Ali, H.; Chali, Y.; Hasan, S. A. (2010): Automation of question generation from sentences. *Proceedings of the Third Workshop on Question Generation*, pp. 58-67.

Bahdanau, D.; Cho, K.; Bengio, Y. (2014): Neural machine translation by jointly learning to align and translate. *Proceedings of the International Conference on Learning Representations*.

Bengio, Y. (2012): Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade*, pp. 437-478.

Bollacker, K. D.; Cook, R. P.; Tufts, P. (2007): Freebase: a shared database of structured general human knowledge. *Proceedings of AAAI Conference on Artificial Intelligence*, pp. 1962-1963.

Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; Yakhnenko, O. (2013): Translating embeddings for modeling multi-relational data. *Proceedings of NIPS*, pp. 2787-2795.

Cai, Q.; Yates, A. (2013): Large-scale semantic parsing via schema matching and lexicon extension. *Proceedings of ACL*, pp. 423-433.

Chen, W.; Aist, G.; Mostow, J. (2009): Generating questions automatically from informational text. *Proceedings of the 2nd Workshop on Question Generation*.

Chung, J.; Ãaglar GülÃ§ehre; Cho, K.; Bengio, Y. (2014): Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555*.

Duma, D.; Klein, E. (2013): Generating natural language from linked data: unsupervised template extraction. *IWCS*, pp. 61-63.

Graesser, A. C.; Gordon, S. E.; Brainerd, L. E. (1992): QUEST: a model of question answering. *Computers & Mathematics with Applications*, vol. 23, no. 6, pp. 733-745.

Greff, K.; Srivastava, R. K.; Koutník, J.; Steunebrink, B. R.; Schmidhuber, J. (2017): LSTM: a search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, pp. 2222-2232.

Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; Bizer, C. (2015): Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, vol. 6, no. 2, pp. 167-195.

Kingma, D.; Ba, J. (2015): Adam: a method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations.*

Li, J.; Monroe, W.; Shi, T.; Ritter, A.; Jurafsky, D. (2017): Adversarial learning for neural dialogue generation. *Proceedings of EMNLP*.

Lopez, V.; Unger, C.; Cimiano, P.; Motta, E. (2013): Evaluating question answering overlinked data. *Web Semantics Science Services Agents on the World Wide Web*, vol. 21, no. 4, pp. 3-13.

Luong, M. T.; Sutskever, I.; Le, Q. V.; Vinyals, O.; Zaremba, W. (2014): Addressing the rare word problem in neural machine translation. *Bulletin of University of Agricultural Sciences and Veterinary Medicine Cluj-Napoca. Veterinary Medicine*, vol. 27, no. 2, pp. 82-86.

Mannem, P.; Prasad, R.; Joshi, A. (2010): Question generation from paragraphs at UPenn: QGSTEC system description. *Proceedings of the Third Workshop on Question Generation*.

Mendes, A. C.; Piwek, P.; Boyer, K. E. (2012): Question generation based on lexico-syntactic patterns learned from the web. *Dialogue & Discourse*, vol. 3, no. 2.

Neco, R. P.; Forcada, M. L. (1997): Asynchronous translations with recurrent neural nets. *International Conference on IEEE*, vol. 4, pp. 2535-2540.

Olney, A. M.; Graesser, A. C.; Person, N. K. (2012): Question generation from concept maps. *Dialogue & Discourse*, vol. 3, no. 2, pp. 75-99.

Papineni, K.; Roukos, S.; Ward, T.; Zhu, W. J. (2002): BLEU: a method for automatic evaluation of machine translation. *Proceedings of ACL*, pp. 311-318.

Petrochuk, M.; Zettlemoyer, L. (2018): Simple questions nearly solved: A new upper-bound and baseline approach. *Proceedings of EMNLP.*

Satanjeev, B. (2005): Meteor: an automatic metric for mt evaluation with improved correlation with human judgments. *Proceedings of ACL*, pp. 228-231.

Serban, I. V.; GarcÃadurÃ¡n, A.; Gulcehre, C.; Ahn, S.; Chandar, S. et al. (2016):

Generating factoid questions with recurrent neural networks: the 30 m factoid question-answer corpus. *Proceedings of ACL*, vol. 1, pp. 588-598.

Su, Y.; Sun, H.; Sadler, B.; Srivatsa, M.; Gur, I. et al. (2016): On generating characteristic-rich question sets for QA evaluation. *Proceedings of EMNLP*, pp. 562-572.

Sutskever, I.; Vinyals, O.; Le, Q. V. (2014): Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, pp. 3104-3112.

Yao, X. (2010): Question generation with minimal recursion semantics. *Proceedings of the Third Wokshop on Question Generation.*

Yih, W. T.; Chang, M. W.; He, X.; Gao, J. (2015): Semantic parsing via staged query graph generation: question answering with knowledge base. *Proceedings of ACL*, pp. 1321-1331.

Yu, L.; Zhang, W.; Wang, J.; Yu, Y. (2017): SeqGAN: sequence generative adversarial nets with policy gradient. *Proceedings of AAAI Conference on Artificial Intelligence*.

Zeng, D.; Dai, Y.; Li, F.; Sherratt, S.; Wang, J. (2018): Adversarial learning for distant supervised relation extraction. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 121-136.