# Self-Attention Architectures for Answer-Agnostic Neural Question Generation

**Thomas Scialom**
LIP6 - Sorbonne Universités
reciTAL
thomas@recital.ai

**Benjamin Piwowarski**
CNRS
LIP6 - Sorbonne Universités
UPMC Univ Paris 06 UMR 7606
Benjamin.Piwowarski@lip6.fr

**Jacopo Staiano**
reciTAL
jacopo@recital.ai

## Abstract

Neural architectures based on self-attention, such as Transformers, recently attracted interest from the research community, and obtained significant improvements over the state of the art in several tasks. We explore how Transformers can be adapted to the task of Neural Question Generation without constraining the model to focus on a specific answer passage. We study the effect of several strategies to deal with out-of-vocabulary words such as copy mechanisms, placeholders, and contextual word embeddings.

We report improvements obtained over the state-of-the-art on the SQuAD dataset according to automated metrics (BLEU, ROUGE), as well as qualitative human assessments of the system outputs.

## 1 Introduction

The Machine Reading Comprehension (MRC) community focuses on the development of models and algorithms allowing machines to correctly represent the meaning imbued in natural sentences, in order to perform useful and valuable high-level downstream tasks such as providing answers to questions, generate summaries, and generate relevant questions given a piece of text. Performance on those downstream tasks is indicative of the extent to which the different proposed architectures are able to capture meaning from natural language input.

Recently, neural architectures based on self-attention have obtained significant improvements over the state of the art in several tasks such as language modelling and machine translation, for which abundant data is available. Yet, they have not been thoroughly evaluated on problems for which relatively scarcer datasets are available. We thus investigate the application of Transformers to the task of Neural Question Generation (NQG): given a text snippet, the model is called to generate relevant and meaningful questions about it.

本文针对的是答案不可知的 QG任务.

Question Generation (QG) is an active field of research within the context of machine reading. it matches human behavior when assessing comprehension on a given topic: an expert is able to ask the relevant questions to others to assess their competences. Its potential applications cover a broad range of scenarios, such as Information Retrieval, chat-bots, AI-supported learning technologies. Furthermore, it can be used as a strategy for data augmentation in the context of Question Answering systems.

The QG task has been originally tackled using rule-based systems (Rus et al., 2010), with the research community turning to neural approaches in recent years. In its most popular declination, the task is *answer-aware*, i.e. the target answer within the source text is known and given as input to the QG model (Zhou et al., 2017). Under this scenario, Song et al. (2017) proposed a generative model, jointly trained for question generation and answering. More recently, Zhao et al. (2018) obtained state-of-the-art results using a gated self-attention encoder and a maxout pointer decoder. All these works employ the SQuAD (Rajpurkar et al., 2016) Question Answering dataset, thus directly leveraging the provided answer spans. Conversely, the *answer-agnostic* scenario lifts the constraint of knowing the target answers before generating the questions; Du et al. (2017) proposed an end-to-end sequence to sequence approach, based on a RNN encoder-decoder architecture with a global attention mechanism.

While casting NQG as *answer-aware* is certainly relevant and useful (for instance, as a data-augmentation strategy for question answering data), the ability of generating questions without such constraint is very attractive. Indeed, removing the dependency on an *answer-selection*

*Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6027–6032
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

component allows to reduce the bias towards named entities, thus increasing the model's degrees of freedom. This makes the task more challenging, but potentially more useful for certain applications – e.g. those requiring a natural interaction with a final user. In this work we follow the task as originally defined by Du et al. (2017): we avoid constraining the generation based on a specific answer, effectively operating in an end-to-end *answer-agnostic* scenario.

To adapt Transformers to the NQG task, we complement the base architecture with a copying mechanism, placeholders, and contextual word embeddings: those mechanisms are useful for the treatment of out-of-vocabulary words, which are more likely to affect performance in data-scarce tasks. We study the effect of each of those mechanisms on architectures based on self-attention, reporting improvements over the state-of-the-art systems.

## 2 Architecture

Neural sequence-to-sequence models often rely on Encoder-Decoder architectures: indeed, Recurrent Neural Networks (RNNs) have consistently provided state-of-the-art results for Natural Language Processing tasks such as summarization (Chopra et al., 2016) and translation (Sutskever et al., 2014). Drawbacks of RNN models include the inherent obstacles to parallelism and the consequent computational cost as well as the difficulties in handling long-range dependencies. The recently proposed Transformer model (Vaswani et al., 2017) has proved to be very effective on several tasks (Devlin et al., 2018; Radford et al., 2018), overcoming such issues by not relying on any recurrent gate: it can be briefly described as a sequence-to-sequence model with a symmetric encoder and decoder based on a self-attention mechanism. For an exhaustive description, we refer the reader to (Vaswani et al., 2017) or high-quality blog posts (e.g. "The annotated Transformer"[1]).

模型

Implementation-wise, we used a smaller architecture, with the following hyper-parameters: N = 2 (number of blocks), d_model = 256 (hidden state dimension), d_ff = 512 (position-wise feed-forward networks dimension), h = 2 (number of attention heads). Experiments run with the original hyper-parameters as proposed by Vaswani et al. (2017)[2] obtained consistent and numerically similar results. Throughout our experiments, we used the `spaCy 2.0` library[3] for Named Entity Recognition (NER), Part-of-Speech (POS) tagging, and tokenization.

## 3 Experiments

In a preliminary experiment, we observed poor performances when applying a Vanilla Transformer architecture to the NQG task: we thus investigate how several mechanisms can be exploited within a Transformer architecture and how they affect the performances on the task. In the following, we describe and evaluate the benefits of augmenting the base Transformer architecture with: 只用普通的transformer 表现不佳.

- a copying mechanism;
- a placeholding strategy;
- and, contextualized word embeddings.

### 3.1 Data

We resort to the widely used Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016): it contains roughly 100,000 questions posed by crowd-workers on selected Wikipedia articles; each question is associated with the corresponding answer, and with the reading passage (the *context*) that contains it. In our experiments, we only use the question-context pairs. (不考虑 answer)

We evaluate performances through the commonly used BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), and compare with the current state-of-the-art *answer-agnostic* NQG model described in (Du et al., 2017), considering the question context at sentence-level and using exactly the same splits provided by the authors[4].

### 3.2 Context-free Word Representations

To deal with rare/unseen words, the Transformer (Vaswani et al., 2017) architecture leverages large amounts of data and sub-word tokenization; in Table 1 we show how the performance obtained with a Vanilla Transformer is not satisfactory on the NQG task.

[1] `http://nlp.seas.harvard.edu/2018/04/03/attention.html`

[2] N=6, d_model=512, d_ff=2048, h=8.

[3] `http://spacy.io`

[4] `https://github.com/xinyadu/nqg/tree/master/data/raw`

都建立在 Transformer_base 上

| | **BLEU1** | **BLEU2** | **BLEU3** | **BLEU4** | **ROUGE-L** | *copy%* |
|---|---|---|---|---|---|---|
| *Vanilla Transformer*(使用sub-word token) | 36.13 | 17.77 | 10.04 | 6.04 | 33.17 | *4.2* |
| *Transformer_base*(使用预训练的glove + pos Embedding) | 38.74 | 20.54 | 12.26 | 7.66 | 35.69 | *5.7* |
| *+Copying* | 39.81 | 22.47 | 14.25 | 9.32 | 37.28 | *9.1* |
| *+ELMO* | 40.44 | 23.87 | 15.74 | 10.62 | 38.32 | *6.5* |
| *+Copying+ELMO* | 41.72 | 25.07 | 16.77 | 11.58 | 39.22 | *10.4* |
| *+Placeholding* | 41.54 | 25.52 | 17.56 | 12.49 | 39.26 | *48.4* |
| *+Placeholding+ELMO* | 42.2 | 26.2 | 18.14 | 12.92 | **40.23** | *49.4* |
| *+Placeholding+Copying* | 42.72 | **26.52** | 18.28 | 13.0 | 39.63 | *50.9* |
| *+Placeholding+Copying+ELMO* | **43.33** | 26.27 | **18.32** | **13.23** | **40.22** | *51.7* |
| Du et al. (2017) | 43.09 | 25.96 | 17.50 | 12.28 | 39.75 | - |

弥补有些命名实体没有被正确识别的情况.

弥补命名实体的语义表达

Table 1: Comparison with SOTA; the last column reports the percentage of OOV/placeholders tokens propagated correctly (according to the ground truth) from the source contexts to the generated questions. To assess model stability, we independently trained 10 models with our best architecture, and computed the standard deviation of their BLUE4 performances on the test set: $std < 0.009$.

We hypothesize that this is a consequence of the relatively small size of the task-specific data. Therefore, in our experiments, we use word-level tokenization and GloVe (Pennington et al., 2014) as context-free pre-trained word vectors[5].

Further, consistently with (Chen and Manning, 2014; Zhou et al., 2017), we augment the word representation using learned POS embeddings.

The *Transformer_base* architecture, upon which all subsequent models are built, uses word-level tokenization and pre-trained GloVe embeddings instead of sub-word tokenization as in the *Vanilla Transformer*.

### 3.3 Placeholding Strategy

One method to help the model deal with rare/unseen words is to replace specific tokens with fixed placeholder keywords. Such mechanism is often used in industry-grade Neural Machine Translation systems (Crego et al., 2016; Levin et al., 2017), to enforce the copy of named entities from the source to the target language.

假设：命名实体更容易是 OOV.

Recognizing that named entities are also likely to be among rare/unseen tokens, we resort to such strategy and replace them with fixed tokens: all tokens in the context that are marked as named entity by the NER model are replaced with a token indicating their entity type and order of appearance, with the mapping kept in memory.

For instance, "*Nikola Tesla was born in 1856.*" becomes "*Person_1 Person_2 was born in Date_1*". At training time, the same procedure is applied to the target questions; at inference time, the placeholders are replaced by the corresponding named entities as a post-processing step. This means that a different, randomly initialized, learnable vector is used as embedding for each placeholder, in place of the GloVe representation corresponding to the original token (or to OOV).

As shown in Table 1, this mechanism alone allows the *Transformer_base* architecture to achieve state-of-the-art results. Further, it provides the biggest relative improvement wrt the base architecture. This can be explained by the nature of the SQuAD dataset, in which more than 50% of the answers are named entities (see Table 2 in Rajpurkar et al. (2016)), consistently with the percentage of tokens copied by the placeholding mechanism alone. Moreover, placeholding allows for a significant reduction of the vocabulary size (∼30%).

Nonetheless, a strong limitation of placeholding lies in its full dependency on the NER tagger: if the latter fails to recognize an entity, placeholding has no effect – which is especially damaging when a word was not frequent enough to be included in the vocabulary.

### 3.4 Copying Mechanism

As the questions generated from a given context usually tend to refer to specific phrasing or entities appearing therein, Gulcehre et al. (2016) propose using a pointing mechanism (called *pointer-softmax*) to select words to be copied from the source sentence; intuitively, such method is of particular use in the case of rare or unknown words.

[5] http://nlp.stanford.edu/data/glove.840B.300d.zip

|  | Correctness | Fluency | 语义有效<br>Soundness | Answerability | Relevance |
|---|---|---|---|---|---|
| *Transformer_base* | 4.49 | 4.02 | 3.33 | 1.7 | 2.51 |
| *+Placeholding+Copying+ELMO* | 4.5 | 4.12 | 3.78 | 2.87** | 3.59* |
| Du et al. (2017) | 4.53 | 4.15 | 3.64 | 2.45 | 3.27 |

Table 2: Human assessment: two-tailed t-test results are reported for our best method compared to Du et al. (2017) ($* : p < 0.05$, $** : p < 0.005$).

The generation probability $p_{gen} \in [0, 1]$ at timestep $t$ is calculated as:

attention后的context vector

decoder状态

$$p_{gen} = \sigma(W \cdot (h^* \oplus s_t \oplus x_t))$$

decoder输入

where $W$ is a learnable parameter vector, $h^*$ represents the context and is computed through attention (*i.e.* as a linear combination of the final encoder representations $[h_1, \ldots, h_t]$), $s_t$ is the decoder state, and $x_t$ the decoder input. We tested several attention mechanisms to enable the copying, including global attention (Luong et al., 2015); since no significant differences were observed, for our experiments we used the raw attention scores of the Transformer, thus avoiding the addition of more trainable parameters.

The results reported in Table 1 show how the addition of copying benefits the model performance, and particularly how it allows the amount of tokens copied to increase, complementing the placeholding mechanisms when the named entities are not correctly recognized. The following example from SQuAD exemplifies the contribution of the copying mechanism: given the context *"Beyoncé attended St. Mary's elementary school in Fredericksburg, Texas, where [...]"*, for which the NER fails to mark *Beyoncé* as named entity (moreover, *Beyoncé* is not in the vocabulary) the Transformer + placeholding produces *where did madonna attend st. mary 's school ?*, while the addition of copying allows to correctly recover the correct entity and allows the model to emit a correct question: *where did beyoncé attend school ?*

### 3.5 Contextualized Embeddings

Contextualized representation approaches allow to compute the embedding of a given token depending on the context it appears in, as opposed to the fixed, context-free vectors provided by GloVe, therefore 上下文感知的Embedding allowing to capture more information for OOV tokens. The placeholding strategy described above has the downside 负面 of depriving 剥夺 the input text representation of any semantic information besides the entity type. For instance, two entities such as Tesla and Edison could have close representations in the word embedding space, within a scientific-related subset of tokens: the use of a placeholder would thus prevent the use of such information. Therefore, we concatenate the context-free vectors (see 3.2) for a specific token with the corresponding ELMO (Peters et al., 2018) representation at the encoding stage. In our experiments, those are only used in the encoding stage since they can only have a meaning when applied to full sentences.

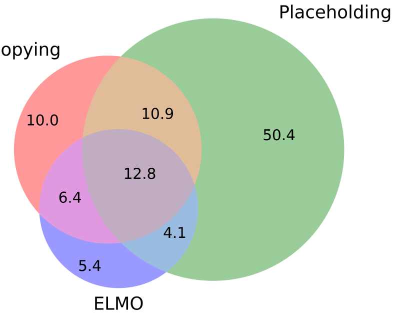

Figure 1: Percentage of OOV tokens copied by the different mechanisms and combinations thereof, over all OOV tokens copied.

Combined with the previously described mechanism, contextualized embeddings allow to further improve the performances, obtaining a BLEU4 score of 13.23, almost one absolute point above the current state-of-the-art in the *answer-agnostic* task. As depicted in Figure 1, they also contribute to the selection of relevant OOV tokens to copy from the context to the generated question.

## 4 Human Assessment

Finally, we proceeded to a qualitative evaluation of the generated outputs, by randomly sampling 100 context-question pairs from the test set. Three professional English speakers were asked to evaluate, the questions generated by: a) *Transformer_base*, b) our best performing model, and c) the state-of-the-art model by Du et al. (2017)[6].

[6] To reproduce the outputs of Du et al. (2017) we used the code from `https://github.com/xinyadu/nqg`.

The questions generated by the different models were shuffled before the assessment. Ratings were collected on a 1-to-5 likert scale, to measure to what extent the generated questions were:

- answerable, by looking at their context (*Answerability*);
- relevant to their context (*Relevance*);
- grammatically correct (*Correctness*);
- semantically sound (*Soundness*);
- and, well-posed and natural (*Fluency*).

As shown in Table 2, this analysis indicates a qualitative improvement over SOTA in terms of *Relevance* and *Answerability*.

Below, we report a few sample outputs, randomly selected, generated by the proposed model.

## 5 Conclusions and Future Work

We have described a preliminary study on the adaptation of Transformer architectures to Neural Question Generation. The results obtained show the contribution of auxiliary techniques such as copying mechanism, placeholding, and contextualized embeddings, which complement each other. The best performance is obtained when using the three mechanisms altogether, reaching an improvement of almost one BLEU4 point (and of 0.5 for ROUGE-L) over the current state-of-the-art approaches. Furthermore, a qualitative assessment indicated improvements in terms of *Relevance* and *Answerability*.

We are extending the proposed approach to other QA datasets, and adapting it to use pretrained language models such as BERT (Devlin et al., 2018), to evaluate the consistency of the mechanisms introduced.

**Sentence 1:** Under Rockne, the Irish would post a record of 105 wins, 12 losses, and five ties.
**Human:** What was the amount of wins Knute Rockne attained at Notre Dame while head coach?
**Ours:** how many losses did the irish have ?

**Sentence 2:** Chopin was of slight build, and even in early childhood was prone to illnesses.
**Human:** What was Frédéric prone to during early childhood as a result of his slight build?
**Ours:** what type of disease did chopin have ?

**Sentence 3:** Montana contains thousands of named rivers and creeks, 450 miles (720 km) of which are known for ”blue-ribbon” trout fishing.
**Human:** How many miles of rivers are known for high class trout?
**Ours:** how many miles of rivers does montana contain ?

**Sentence 4:** In 1648 before the term genocide had been coined, the Peace of Westphalia was established to protect ethnic, national, racial and in some instances religious groups.
**Human:** What year was the Peace of Westphalia signed?
**Ours:** when was the peace of westphalia established ?

**Sentence 5:** A bactericidal activity of antibacterials may depend on the bacterial growth phase, and it often requires ongoing metabolic activity and division of bacterial cells.
**Human:** What besides ongoing metabolic activity is required in bactericidal activity?
**Ours:** what type of activity do antibacterials depend on ?

**Sentence 6:** The Montana Act led to the arrest of over 200 individuals and the conviction of 78, mostly of German or Austrian descent.
**Human:** How many people were arrested from the Montana Act?
**Ours:** how many individuals were killed in the montana act ?

## References

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, et al. 2016. Systran's pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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 (Volume 1: Long Papers)*, volume 1, pages 1342–1352.

Caglar Gulcehre, 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 (Volume 1: Long Papers)*, volume 1, pages 140–149.

Pavel Levin, Nishikant Dhanuka, Talaat Khalil, Fedor Kovalev, and Maxim Khalilov. 2017. Toward a full-scale neural machine translation in production: the booking. com use case. *arXiv preprint arXiv:1709.05820*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

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 on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pretraining. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_ understanding_paper. pdf*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

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*, INLG '10, pages 251–257, Stroudsburg, PA, USA. Association for Computational Linguistics.

Linfeng Song, Zhiguo Wang, and Wael Hamza. 2017. A unified query-based generative model for question generation and question answering. *arXiv preprint arXiv:1709.01058*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proc. NIPS*, Montreal, CA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.