| Model | Question type | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | what | how | when | which | where | who | why | yes/no |
| seq2seq+AP | 77.3% | 56.2% | 19.4% | 3.4% | 12.1% | 36.7% | 23.7% | 5.3% |
| ASs2s | 82.0% | 74.1% | 43.3% | 6.1% | 46.3% | 67.8% | 28.5% | 6.2% |

Table 3: Recall of interrogative word prediction.

answer-separated seq2seq without the answer-separated decoder but with a general LSTM decoder.

As shown in Table 1, ASs2s outperforms all of the previous NQG models on both data splits by a great margin, showing that separate utilization of target answer information plays an important role in generating the intended questions. With the help of answer-separated decoder, ASs2s-<a> still outperforms the previous NQG models except for ROUGE-L on data split-1. However, there is a considerable decrease in all metrics compared to the complete model. This results from the fact that answer separation prevents generated question from including the answer. Similarly, ASs2s-keyword has a big drop in performance and this verifies that the keyword-net has actual impact on improving the performance. ASs2s-ASdec has greater decrease in all metrics compared to the ASs2s. This is a very natural result because without the answer-separated decoder, the model has to generate questions by only relying on context around the target answer position without knowledge of the target answer.

## Impact of Answer Separation

Answer separation helps the model generate the right question for the given target answer. Since the base model does not utilize the target answer information, we further define **seq2seq+AP(Answer Position)** as base model with answer position feature (**?**) for comparison. We show the benefits of answer-separated seq2seq in three aspects.

**Answer Copying Frequency** If a NQG model captures the question target well, the generated question will rarely include the target answer. We verify the assumption by computing the percentage of generated questions including target answers. Since (**?**) ignores the target answer, we choose seq2seq+AP to represent (**?**) with answer position feature. Further, we choose the previous state-of-the-art (**?**) for comparison because both (**?**) and (**?**) use the copy mechanism.

As shown in Table 2, the percentage that the target answers are either completely or partially included in the generated questions is significantly lower in our model. We also figure out an interesting observation: even though (**?**) is the previous state-of-the-art NQG model, it generates more irrelevant questions to the target answer when compared to seq2seq+AP. This observation indicates the negative effect of copy mechanism that the target answer inside the passage is unintentionally copied to the generated question.

**Interrogative Word Prediction** To figure out the effect of answer-separated seq2seq on question type prediction, we compare the recall of each interrogative word prediction between the generated questions of answer-separated seq2seq and seq2seq+AP. We group questions into 8 categories: what, how, when, which, where, who, why and yes/no. As shown in Table 3, answer-separated seq2seq has better recall score over seq2seq+AP in all categories. Especially, the recall of question types how, when, where and who improved in big magnitude. Both model's recall of question type what is very high because what takes up more than half of the whole training set (55.4%). Both model's recall of type which is very low. This may result from the fact that some combinations like which year and which person may be generated as where and who respectively. For question types why and yes/no which only take up 1.5% and 1.2% of the training set respectively, both models did not perform well due to the small amount of data.

**Attention from <a>** We verify the effect of replacing answer with <a> by comparing attention matrices. Given the passage john francis o'hara was elected president of notre dame in 1934. and the target answer john francis o'hara, following Figure 3(a) and Figure 3(c) show the attention matrices produced by our answer-separated seq2seq and seq2seq+AP respectively.

As shown in Figure 3(a), the interrogative word ”who” gets most of the attention weights(higher attention weights) from the <a> token in our answer-separated seq2seq. Further more, Our model can generate a question that is exactly related to the target answer. With additional answer position features as in Figure 3(c), only a part of answer is attended while generating the interrogative word who. In this case, if the answer has some contextual information, then the model may omit it, generating an unintended question. Also, the generated question contains john francis which is a part of the target answer. We infer that the encoder tends to utilize more information from the word embeddings rather than answer position features, since the word embedding has far more information than answer position features.

## Question Generation for Machine Comprehension

By training a machine comprehension system on the synthetic data generated by our model, we verify that our model has an enough ability to generate natural and fluent questions. By changing the position of the <a> token, we can easily produce various questions with our model. Figure 3(a) and Figure 3(b) shows one example where we use our model to generate two different questions corresponding to different target answers from the same input passage.

We experiment with QANet (**?**) on SQuAD dataset to verify whether the generated questions from our model are valid or not. Since most of the answers correspond to named entities, we use words and phrases that are named entities from

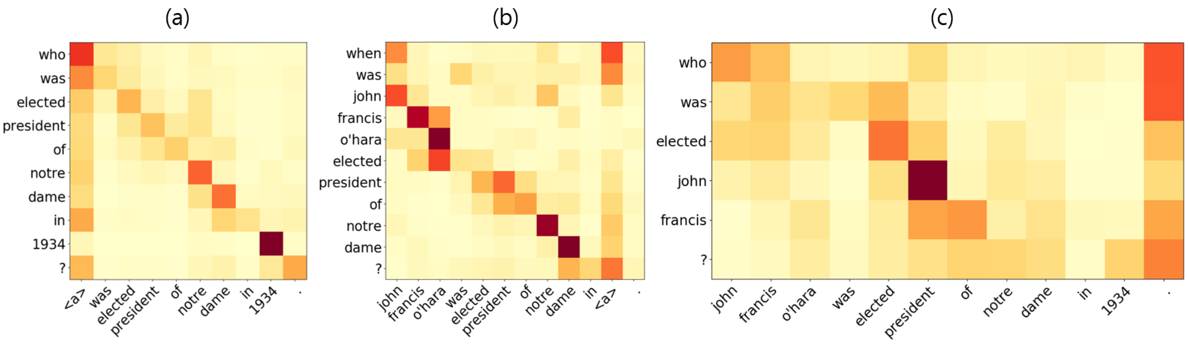

Figure 3: (a) and (b) show attention matrices of our model given a passage with two different target answers. (c) shows an attention matrix of seq2seq+AP given the same passage and the target answer as (a).

| Answers | Exact Match (EM) | F1 score |
|---|---|---|
| ALL | 22.72 | 31.58 |
| NER | 49.09 | 56.57 |

Table 4: Performance of the machine comprehension system which is trained only with synthetic data generated by our NQG model.

training part of data split-1 as target answers. Then, we pair those answers with corresponding passages. We also make sure that selected answers are not overlapped with answers in the original SQuAD dataset because our NQG model is trained with the target answer provided with SQuAD dataset. If answers are overlapped, our model may generates exact the same questions as the golden questions. then we pair those answers with corresponding passages.

To organize the dataset in the same way as SQuAD dataset, (*paragraph*, *question*, *answer position*) triplets, we trace the passage in data split-1 in the original paragraph and re-compute the answer position as well. We finally make a synthetic data with about 50k questions and train the machine comprehension system only with our synthetic data. As shown in Table4, the machine comprehension system achieves EM/F1 score of 22.72/31.58 in public SQuAD dev set. This result is far below the result 68.78/78.56 of the case when the model is trained with the original training set. However, considering our synthetic data only consists of target answers with single named entity, we further check EM/F1 score of partial dev set that only has a single named entity as the answer. We find that in the 10k dev set, about 40 percent of the data has an answer with a single named entity and the machine comprehension system achieves EM/F1 score of 49.09/56.57 with those parts of the data. Since the SQuAD dataset is a human-made dataset, this result sufficiently shows that our answer-separated seq2seq can generate valid questions that can be acceptable both by human and machine comprehension systems.

## Conclusion

In this paper, we investigate the advantages of answer separation in neural question generation. We observe that existing NQG models suffer from a serious problem: a significant proportion of generated questions include words in the question target, resulting in the generation of unintended questions. To overcome this problem, we introduce a novel NQG architecture that treats the passage and the target answer separately to better utilize the information from the both sides. Experimental results show that our model has a strong ability to generate the right question for the target answer in the passage. As a result, it yields a substantial improvement over previous state-of-the-art models.

## Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean government (MSIT) (No. 2016M3C4A7952632), Industrial Strategic Technology Development Program (No. 10073144) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea). Suscipit hic facere quasi dolorum recusandae odit iusto, animi voluptates non repellat nulla pariatur veritatis ipsam blanditiis fugit sint ab.Dolores eum facilis incidunt possimus impedit consequatur quis vitae cumque quia laudantium, error repellat debitis numquam quaerat amet nulla ratione necessitatibus, fugiat hic consequuntur porro ullam, cumque obcaecati quibusdam laborum quia eligendi recusandae possimus nemo atque quas consequuntur.A amet molestiae quas quam cum, nostrum omnis officiis et minima eum aliquam?Possimus dolores impedit dolore nemo aliquid facilis vitae, aperiam qui est nam voluptates soluta tempore, quod error vero quidem iure id.Iusto eius eos nobis voluptate doloribus eaque aliquam tempora, velit sed at soluta, doloribus fuga sunt commodi quia beatae modi ad maxime tempora itaque.Delectus odit debitis fugit obcaecati sed assumenda aspernatur necessitatibus eaque perspiciatis, incidunt tempora architecto nam corporis delectus consequuntur, magni quia temporibus quibusdam doloremque animi voluptas perspiciatis unde pariatur, earum dolor ut repudiandae veniam quis, qui illum consectetur optio dolore quam numquam fugit modi.