Exploring the use of Question Generation for Question-Answering-based Argument Role Labeling

Anonymous ACL submission

Abstract

Argument Role Labeling (ARL), a sub-task of event extraction, aims at identifying and classifying arguments from natural language text. Traditionally, the expense and limitations of collecting annotated event data hinder the use of ARL models in practical situations. To mitigate the shortage of available annotated event data, we propose, starting from the recent QA-based approach to ARL (Du and Cardie, 2020; Liu et al., 2020), a novel data augmentation method that can easily and cheaply enlarge the in-domain data for ARL. First, we convert the ARL task into an equivalent Question-Answering (QA) task. Secondly, we experimented with several data augmentation systems including models for answer extraction and question generation. Experimenting on the ACE dataset for ARL task, we explore the influence of the different parameters on the performance. In particular, we find that even a small quantity of high quality target-related QA pairs can outperform the use of large QA data.[1]

1 Introduction

Argument Role Labeling (ARL) aims at identifying and classifying arguments and roles from a natural language text, given the event types. It is an important task within Information Extraction that is necessary for the understanding of an event as exemplified in Table 1, describing argument roles and spans for a given text. Recent works proposed a Question-Answering (QA) approach to ARL (Du and Cardie, 2020; Liu et al., 2020) in which each argument type is targeted by a question derived from an event ontology. For example, in Table 1, for the "Marry" event type, the Person Argument is targeted by the question "Who is married?". This approach, in which both training and test sets are converted into a QA format, avoids the entity recognition step that usually leads to error propagation (Du and Cardie, 2020), by relying on a QA system.

[1]All the code and paper artifacts will be made available upon publication.

They were married in Spain.	
Person-Arg	The people who are married Who is married?
Place-Arg	Where the marriage takes place Where is someone married?
Time-Arg	When the marriage takes place When is someone married?

Table 1: An example of "Marry" Event for the Argument Role Labeling task and its Question-Answering formulation

As human annotation for ARL is costly and requires expertise, the use of ARL systems in realistic situations is constrained to the limited amount of in-domain training data.

In this paper, we introduce a system that can automatically create in-domain data for an ARL task. Starting from the QA formulation of ARL where training examples are converted into QA pairs, we propose using Question Generation (QG) in order to bootstrap the QA pairs, creating additional in-domain training data. For this purpose, our system involves an answer selection stage and then a question generation stage trained on the existing QA pairs. With unlabeled new text, our system will generate more QA pairs in the same format as our training QA pairs. In comparison with transfer learning approaches that make use of existing QA datasets (Liu et al., 2020; Lyu et al., 2021), our data is closely connected to the specific event types of the task and our system can be adapted to any possible ARL dataset.

We design three data augmentation systems, an Answer Extraction (AE) - QG system, a Semantic Role Labeling (SRL) - QG system and an SRL - predicate-aware QG system. Experimenting on the ACE dataset (Walker et al., 2006) and focusing on scenarios where the answer is in the text, we show that our AE-QG system is the most helpful among the above three systems (AE-QG, SRL-QG,

November 13, 2004, Iranian representatives say negotiations with Europe on its nuclear program are in the final stages.
QA-WikiNews no ACE: When did Iranian representatives say negotiations with Europe are in the final stages?
QA-WikiNews: When is the meeting?

Table 2: The question QG generated is a contextualized questions, while QG-finetuned on ACE generates one of the fixed questions in ACE-QA.

SRL-QG_prd_aware) and that it improves the performance of the QA approach baseline. We also analyze the factors that influence the effectiveness of the dataset from both answer and question perspectives and find that the quality of generated QA pairs in the target format is more important than the quantity.

2 Data

ACE: ACE-2005 (Walker et al., 2006), a human annotated event dataset, incorporates over 33 event types (e.g. Marry, Attack). In each event type, the definitions of the event and arguments are clearly described (Table 1). For English, the dataset includes 535 articles from various of news, broadcasts, dialogues and blogs.

We followed the split of Lin et al. (2020), dividing it into train and test sets. We employed the set of questions from Lyu et al. (2021)[2] to convert the data into a QA format, resulting in ACE-QA-train and ACE-QA-test. For each argument role in an event type, there is a unique fixed question (see examples in Table 1).[3]

WikiNews: We employed the WikiNews dataset proposed by Trani et al. (2014, 2016). It consists of 604 English news articles and can be viewed as an related-domain source text compared to ACE. We generate QA pairs for the WikiNews text by employing our data augmentation systems (see Section 3), resulting in WikiNews-QA.

SQuAD: SQuAD1.1[4] (Rajpurkar et al., 2016) incorporates 80,000 human-labeled answerable QA pairs and paragraphs extracted from Wikipedia articles. It is considered as an out-domain dataset compared to ACE.

Text: April 7, 2014, writer Peaches Geldof was found dead in her home near Wrotham.
AE input: extract answers: April 7, 2014, ...
AE output: Peaches Geldof <sep> Wrotham <sep>
SRL input: ["April" ... "Peaches", "Geldof"... "found", "dead"... "Wrotham", "."]
SRL output: ["11:B-TMP"..."11:B-A1", "11:I-A1"..."[prd]","11:B-A3"..."11:I-LOC",""]
QG input: generate question: ...writer <hl> Peaches Geldof <hl> was...
prd-aware QG input: generate question: ...<hl> Peaches Geldof <hl> was # found # dead...
QG output: Who is killed?
QA input: ...Peache... [SEP] Who is killed?
QA output: Peaches Geldof

Table 3: Examples of the AE, SRL, QG, QA models input and output.

3 Method

3.1 Models

This section introduces each single model in the QA data augmentation systems we propose. The goal of Answer Extraction (AE) and SRL models is to extract appropriate candidate answers for ACE events. The goal of QG model is to generate ACE-related questions. The goal of the QA model is to solve the ARL task by using the QG output as additional training data.

Answer Extraction (AE) Model It is a T5-small model pre-trained on SQuAD. We employed the Answer Extraction work from Chan and Fan (2019)[5]. The input is the text to extract answers from, starting with the "extract answers:" task indicator. The output is a list of extracted answers, separated by a "<sep>" token. (See in Table 3)

We trained the AE model on ACE-QA-train for 10 epochs and predict on raw WikiNews. From the above example in Table 3, we can have a glimpse that the AE model usually can extract the right candidate answer if it exists, however, it may miss some arguments, such as "April 7, 2014" for a "Die" event Time-Argument.

[2]We compare the fixed questions of (Lyu et al., 2021) and the ones presented in (Du and Cardie, 2020). The baselines are 70.25 and 70.10, respectively. We chose the former for the following experiments.

[3]We only include questions for which the answer appears in the text (has-answer questions). This is motivated by a high no-answer score we obtain for the baseline (90.90), inspiring our focus on the improvement of the has-answer ability.

[4]We abbreviate it as SQuAD henceforth

[5]We use for both AE and QG the implementation at https://github.com/patil-suraj/question_generation

Semantic Role Labeling (SRL) Model It is a first-order CRF model for verb predicate trained on PennTreeBank. We directly apply the work of Zhang et al. (2021)[6] on WikiNews without any further training. The input of the model is a piece of text. The output of the model consists in the tokenized sentences and the corresponding SRL role of each token. (See an example in Table 3)

The advantage of the SRL model is that it is more comprehensive comparing to the AE model, if we incorporate enough SRL roles. The shortages are the possibility of adding many wrong candidate answers (regard as false positive for ACE event) and the exclusion of noun predicates. In the example, it ignores the nominal predicate "dead", the true trigger for the "Die" event.

Question Generation (QG) Model It is a T5-small model pre-trained on SQuAD. We adapted the Question Generation work of Chan and Fan (2019) and fine-tuned on ACE-QA-train for 10 epochs. We trained two versions of the QG model, differentiated by their inputs. Both start with a "generate question:" task indicator but the first (QG) only marks an answer in the text, while the second (prd-aware QG) marks both an answer and its predicate.[7] The output is the question generated for the answer. (See an example in Table 3) The prd-aware QG after SRL is a solution for the false mapping between answer and its predicate in QG. This frequently happens when multiple predicates or ACE event triggers exist in a complex sentence.

Question Answering (QA) model We start with RoBERTa-large language model (Liu et al., 2019), and fine-tuned it on our augmented WikiNews-QA data or additional ACE-QA-train data to evaluate the effectiveness of our generated data for ARL. The input is the text and a question about it while the output is an answer. (See in Table 3)

3.2 Systems

This section discusses the combinations of the single models above for data augmentation in ARL.

AE-QG Pipeline Model It is built on the AE and QG models. The answers extracted from the AE model are all treated as candidate answers for the QG model. We include two modifications on that, the first one is a multiple-answer AE (AE-QG multi-ans). The second one is a post-processing step (AE-QG clean) which eliminates all the QA pairs where the questions are not fixed ACE questions or have a very high frequency in ACE-QA-train, in order to balance the QA data.

SRL-QG Pipeline Model SRL-QG and its variant, the SRL-predicate-aware QG pipeline model, consist of the single SRL and QG/prd-awre QG models. We only include the A0,A1,TMP and LOC arguments as the candidate answers to balance between true positive and false positive answers.

We also experiment with the combination of the AE-QG and SRL-QG models, using both sets of generated questions as additional data and with the use of SQuAD instead of QG.

[6]We employ the package at https://github.com/yzhangcs/crfsrl

[7]the answer and predicate are respectively marked in the text with "<hl>" and "#".

System	Data	Wiki	+ACE
Non-QG baseline	-	-	70.25
Non-QG SQuAD	80k	51.94	71.56
AE-QG	8k	**60.91**	**72.05**
AE-QG (no ACE)	8k	47.49	70.07
AE-QG (multi-ans)	14k	46.96	70.71
AE-QG (clean)	3k	42.44	71.13
SRL-QG	30k	45.12	69.83
SRL-QG (prd-aware)	30k	45.20	71.26
AE-QG+SRL-QG (prd-aware)	38k	57.12	70.57

Table 4: Results of QA data augmentation system. The System column presents the data augmentation systems and the Data column shows the number of created QA pairs on WikiNews (except SQuAD). The F1 results on ACE-QA-test after training a QA model on WikiNews-QA and additionally on ACE-QA-train are offered in the third and forth columns.

4 Results and Analysis

In the results presented in Table 4, we first observe that AE-QG significantly improves the performance of the standard QA approach (paired t-test; p=0.028). We also find that AE surpasses SRL for answer selection and that predicate-aware QG works better in matching text and answers to questions in the SRL approach. We conclude that AE and predicate-aware QG create higher quality QA pairs. Furthermore, Table 5 summarizes the properties of the three datasets used as additional training data.

To gain more insights into the meaning of quality, we dive in each model component respectively. In the below experiments, we apply our AE-QG and SRL-QG systems to ACE-QA-test and compare the results with the gold QA pairs. Firstly, we compare

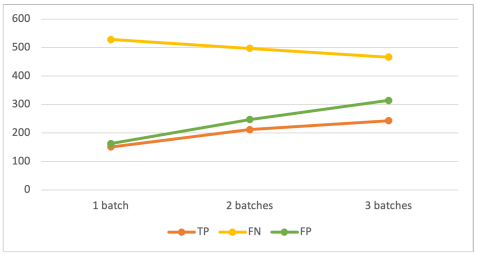

Figure 1: A comparison of AE answers with ACE

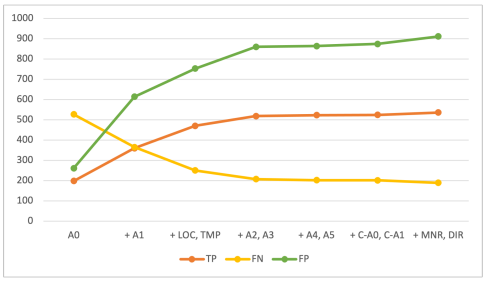

Figure 2: A comparison of SRL roles with ACE

the answers extracted from AE and SRL with the true ACE-QA-test answers. Secondly, assuming the gold answer selection, we compare the question generated from QG and predicate-aware QG with the true questions in ACE-QA-test.

For the first stage, Figure 1 shows the pattern of AE extracted answers as more candidate answers are included. In this approach the number of gold answers missed (FN) is much greater than that of the false answers chosen (FP) and the gold answers identified (TP). With more batches of answers, the FN goes down and the FP goes up faster than TP. Considering the results of AE-QG and AE-QG multi-ans, we infer that FP rate (precision) can be an essential indicator of the data quality, because FP will lead to an error propagation in the QG stage. Figure 2 reveals the connections between true arguments and SRL roles. We see that A0, A1, TMP and LOC play an important role in forming the true ACE arguments. Here, the FP is the highest, almost twice of the TP and the FN is the lowest. Comparing the result of AE-QG and SRL-QG, it verifies again our interpretation of the significance of high precision. Besides, our SRL is a verb-based SRL, and we include all the predicates and arguments identified by the SRL model. Noticeably, a large amount of predicates are not triggers in ARL task, e.g. "see", contributing to a high FP. Examining the ACE event types, we observe that the SRL roles (e.g. A1) are not always event arguments in ACE, producing some FP cases. The FN comes from the shortage of a verb-based SRL which omits the

	SQuAD	**Wiki** (no ACE)	**Wiki**
Size	large	small	small
Domain	Out	Related	Related
Fixed Quest.	No	No	Yes
ACE format	No	No	Yes
Helpful	medium	lowest	highest

Table 5: A summary of the comparisons of the datasets. We inspect the quantity, the source text domain, fix or contextual question, ACE format QA pair or not, and helpful or not as additional data.

nominal predicate's arguments. To improve the accuracy of an answer extraction model, we suggest focusing on a verb and nominal combined SRL approach with a predicate filtering process, e.g. a trigger identification model.

In the question generation stage, given the gold answers, QG generates 338 correct questions out of 726 in total, while predicate-aware QG creates 320 out of 840. The accuracy of QG is higher than predicate-aware QG, which is in contrast with our intuition and the results we got from SRL-QG and SRL-prd-aware QG. Investigating the questions generated by QG and prd-aware QG, we find that most of the questions generated by the two models are same despite labeling the predicate or not. When prd-aware QG marking a non-trigger predicate. we add harmful mark that can bring more confusion to the model and thus, generate wrong questions. However, we also observe cases where the answer is matched to a wrong predicate and question in SRL-QG. This confirms the necessity of a correct predicate mark. Thus, we propose a predicate filtering process, in accordance with our suggestions in the answer selection stage.

5 Conclusion

In this paper, we propose to use Question Generation in order to improve QA-based Argument Role Labeling, designing three novel systems. We show the effectiveness of our approach and perform an in-depth analysis of the different components. The latter shows the importance of generating QA pairs of the same format as the target dataset and of the high TP vs. FP difference in answer selection. Although our AE-QG system achieves the best performance, our analysis shows the potential of the use of SRL for answer selection, which can be further improved by integrating nominal SRL and event type filtering.

Limitations

In our approaches, we only solve the task of Argument Role Labeling, which is a sub-task of Event Extraction. Besides, our training and test data include short paragraphs (less than 3 sentences), and the extension to long paragraphs is not addressed in this paper. In addition, our system is based on the AE-QG or SRL-QG pipeline model, which can be more complicated to train and deploy than the baseline QA system.

Ethics Statement

Our task training data and additional text are all focusing on formal English, news articles etc. We did not explore the effectiveness of our systems on informal language, such as oral communication. Besides, our generated data should only be used for the improvement of the QA ARL task. The usefulness and effects of this data for other purposes was not explored in this paper.

References

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent bert-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on EmpiricalMethods in Natural Language Processing (EMNLP)*, pages 671–683.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332.

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 (EMNLP)"*, pages 2383–2392.

Salvatore Trani, Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, and Raffaele Perego. 2014. Manual annotation of semi-structured documents for entity-linking. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 2075–2077.

Salvatore Trani, Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, and Raffaele Perego. 2016. Sel: A unified algorithm for entity linking and saliency detection. In *Proceedings of the 2016 ACM Symposium on Document Engineering*, pages 85–94.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. Linguistic Data Consortium, Philadelphia, 57.

Yu Zhang, Qingrong Xia, Shilin Zhou, Yong Jiang, Zhenghua Li, Guohong Fu, and Min Zhang. 2021. Semantic role labeling as dependency parsing: Exploring latent tree structures inside arguments. *arXiv preprint arXiv:2110.06865*.

Appendix A: Settings of Five WikiNews-QA datasets

The first one (WikiNews-QA no ACE) was directly generated from the AE-QG system of Chan and Fan (2019) without fine-tunning on ACE. It contains 8060 QA pairs. The second one (WikiNews-QA fine-tuned on ACE) was generated from fine-tuned AE-QG system on ACE-QA-train dataset before predicted on WikiNews. It consists of 8080 QA pairs. (see examples in Table 2) The third one (WikiNews-QA fine-tuned on ACE with multi-answers) was a modified version of Chan and Fan (2019)'s work. It enables multi-answers extraction from a single sentence. This version contains 14427 QA pairs. The forth one (QA-WikiNews fine-tuned on ACE, SRL-QG) employed an SRL model from Zhang et al. (2021) instead of an AE model for the answer selection. When limiting the SRL to A0,A1,TMP (Time) and LOC (Location), more than 30k QA pairs were created. The fifth one (QA-WikiNews fine-tuned on ACE, SRL-predicate-aware QG), based on the forth setting (SRL-QG), re-trained the QG model with a marked predicate as the input. It contains more than 30k QA pairs.