Learning with Limited Data for Multilingual Reading Comprehension

Kyungjae Lee[1*] **Sunghyun Park**[1*] **Hojae Han**[1] **Jinyoung Yeo**[3]
Seung-won Hwang[1†] **Juho Lee**[2]
[1]Yonsei University, [2] NAVER Corp, [3] SK T-Brain

Abstract

This paper studies the problem of supporting question answering in a new language with limited training resources. As an extreme scenario, when no such resource exists, one can (1) transfer labels from another language, and (2) generate labels from unlabeled data, using translator and automatic labeling function respectively. However, these approaches inevitably introduce noises to the training data, due to translation or generation errors, which require a judicious use of data with varying confidence. To address this challenge, we propose a weakly-supervised framework that quantifies such noises from automatically generated labels, to deemphasize or fix noisy data in training. On reading comprehension task, we demonstrate the effectiveness of our model on low-resource languages with varying similarity to English, namely, Korean and French.

1 Introduction

Reading comprehension question answering (RCQA) is one of many well-known NLP tasks, to answer questions based on text understanding. For RCQA, a well-known resource is SQuAD (Rajpurkar et al., 2016) with 100K QA data created by human, followed by NarrativeQA (Kočiskỳ et al., 2018), SQuAD 2.0 (Rajpurkar et al., 2018), and CoQA (Reddy et al., 2018). However, as these datasets support only English, supporting other languages requires either annotation efforts in a comparable scale (Lim et al., 2018), or modeling efforts to overcome the limitation of training resources in terms of **quantity** or **quality**. Our work pursues the latter goal.

To illustrate, consider an extreme scenario of bootstrapping a RCQA model for a new language with no labelled resource. We can overcome the limitation in **quantity** by generating alternative low-quality resources. First, neural machine translation (NMT) can convert existing English annotations into a target language (Faruqui and Kumar, 2015; Ture and Boschee, 2016). For example, (passage p, question q, answer a) in SQuAD can be translated into (p', q', a') in a target language (Asai et al., 2018; Lee et al., 2018). Second, automatic labeling function (Du and Cardie, 2018; Serban et al., 2016) can be adopted to generate synthetic labels in a target language. For example, Du and Cardie (2018) leverages Question Generator (QG) and Answer Extractor (AE) as labeling function for RCQA, to create training resources of a virtually infinite amount from unlabeled corpora.

However, overcoming the quantity limitation using the above methods, leads to **quality** problems. As a consequence, some of the following assumptions on quality may cease to hold, after errors introduced from generation. (1) **Answerability**: the semantic of the translated passage or generated question may shift, so that a question gets unanswerable after generation, or (2) **Answer alignment**: the answer span in a target language may become incorrect. Our goal is estimating the quality of resources and even improving the quality, to overcome the limited quality of training resources.

For the goal, we exploit noisy data from the above two sources, translator and labeling function, considered as **weak labels**, and pursue robust learning overcoming noises: Our key contribution is ***Refinery*** network that predict confidence – quality of (p, q, a) instance into a score in the range of 0 to 1. While most existing weakly supervision approach focuses on generating positive examples, we leverage synthetic negative examples for training *Refinery*, optimized to distinguish positive from negative one.

As shown in Figure 1, a training procedure of

*First two authors equally contributed to this work.
†correspond to seungwonh@yonsei.ac.kr

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 2840–2850, Hong Kong, China, November 3–7, 2019.

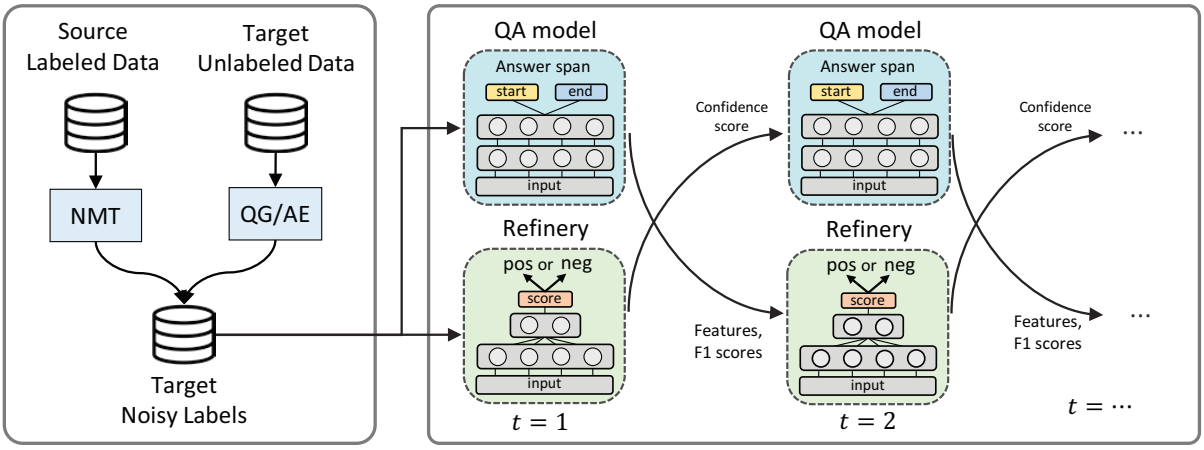


Figure 1: Overview of our approach

QA and *Refinery* is iterative: QA model is used for confidence estimation and *Refinery* is used to determine which noisy instances should be deemphasized in training, or whether to modify wrong labels, towards the direction of improving QA model. As our framework does not re-train or access data generation models, our approach is agnostic to generation model, such that it can extend to generation models not discussed here.

For experiments on RCQA in a new language, our model is evaluated on three human-annotated sets. Our experiments show that our method with *Refinery* augments training data and outperforms all state-of-the-arts. This observation can be generalized over similar and distant language pairs (English with French and Korean respectively), and with and without significant topic overlaps: To validate this claim, we evaluate over widely adopted human generated evaluation sets on Korean and French Wikipedia documents. In addition, our reported gains are orthogonal to a pre-trained model, such that we can easily leverage the strength of the latest BERT model (Devlin et al., 2018), and contribute additional gains. Our implementation is available at: https://github.com/lkj0509/multilingual_MRC.

2 Preliminaries

RCQA problem is defined based on SQuAD dataset (Rajpurkar et al., 2016) in English, which consists of 23,215 passages and 100k+ question-answer pairs requiring the understanding of corresponding passages to answer correctly. Let p be a passage with m words, *i.e.*, $p = \{w_1^p, w_2^p, ..., w_m^p\}$, and q be a question with n words, *i.e.*, $q = \{w_1^q, w_2^q, ..., w_n^q\}$. Then, given a pair of passage p and question q, the objective of RCQA is to estimate a consecutive answer span a, *i.e.*, $a = \{w_i^p, w_{i+1}^p, ..., w_j^p\}$ where $a \subseteq p$. For task evaluation, the estimated answer span a is compared with the ground truth answer span a^* in terms of F1 score at word-level.

The majority of RCQA models (Seo et al., 2016; Yu et al., 2018; Devlin et al., 2018) consist of (a) encoding the passage and question into a fixed-size vector, then (b) decoding to predict the probability of each position in the passage being the start or end of an answer span. As a basic QA model, we start from BiDAF model (Seo et al., 2016), which is a widely known open-source code.[1] BiDAF model uses bi-directional LSTM networks with attention mechanism to align question with passage and vice-versa. More specifically, the probabilities of the starting and ending position are modeled as:

$$\begin{aligned} \mathbf{P}^1 &= softmax(h_1 \cdot [M^0 ; M^1]) \\ \mathbf{P}^2 &= softmax(h_2 \cdot [M^0 ; M^2]) \end{aligned} \quad (1)$$

where h_1, h_2 are trainable weights, and M^0, M^1, M^2 are the hidden states at each LSTM layer to represent the passage words according to the given question. The probability of s-to-e span of the passage is defined as follows:

$$P(a = \{w_s^p, ..., w_e^p\} | p, q) = \mathbf{P}_s^1 \times \mathbf{P}_e^2 \quad (2)$$

In test, the model selects the answer span with the highest probability (Eq. (2)) during post-processing.

[1] https://github.com/allenai/bi-att-flow

3 Data Generation for New Language

This section introduces how we generate weak supervisions, from machine translation and synthetic label generation.

3.1 Method I: Neural Machine Translation

SQuAD is a set of (p, q, a) triple, where answer a to question q can be found as a consecutive substring match in the passage p (*i.e.*, $a \subseteq p$). Due to this property of SQuAD, translating the triple (p, q, a) is non-trivial: when p and a are translated into target language, p_t and a_t, $a_t \subseteq p_t$ may no longer hold. Therefore, we need to find the answer span a_t in the target language, by matching with a_s in the source language s. We overviewed a high-precision baseline (Lee et al., 2018) where a_s and p_s are independently translated, and a_t is found only when the translation of a_s is exactly and consecutively found in that of p_s. However, this suffers low-recall, considering we found only 53.6% spans among whole translated Q-A pairs by the high-precision method.

To complement, we propose a perfect-recall alignment to find 46.4% spans that cannot be extracted by the above method. Our method is extensible to any language with an existing open-source NMT, leveraging only its attention scores (Bahdanau et al., 2014).

(1) One-to-one alignment: A widely adopted simplifying assumption for machine translation is that each target word is aligned to one source language word (Brown et al., 1993). Based on attention score in NMT, each word in a_s can be aligned with word in the target language, by selecting highest score. After 1-to-1 matching, we select the longest sub-sequence as answer span a_t.

(2) Span-to-span alignment: One-to-one assumption is simplifying, by treating values in the alignment matrix as binary and excluding a possibility that a word can be aligned with multiple target words. Instead, we align span-to-span, to calculate a "soft" score between a_s and a_t, and change the span boundary dynamically. To illustrate, assume $a_t = \{i, ..., j\}$ is the answer span found from the one-to-one alignment. Before finalizing this answer, we may ask whether changing the boundary i and j improves the match[2]. Formally, when $S_{m,n}$ denotes (m, n)-th element in the alignment matrix, we can evaluate the match score of between a_s and a_t by averaging all pairwise combinations in the two spans: $\boldsymbol{S} = average(S_{m,n})$ for $\forall m \in a_s$, $\forall n \in a_t$. If changing the target boundary $a'_t = \{i', ..., j'\}$ improves this score, we will make a modification. Specifically, we consider updating the end position to j' in the range of $j \pm N$, within a pre-defined window size N, and enqueue a possible update j' if the average score of $\boldsymbol{S}$ for a_s and a'_t is higher than that of a_s and a_t. Similarly, we compare the scores of start position in the range of $i \pm N$ to enqueue a possible update i'. After this, highest-scoring update pair (i', j') in the queue (Hwang and Chang, 2005) can be aligned.

[2]We notice that 16% spans from 1-to-1 alignment can be changed to gain F1 score of 1% point on Korean test set.

3.2 Method II: Synthetic Label Generation

This section introduces an automatic labeler which can generate question-answer pairs from unlabeled text crawled for the target language. Training data collected via NMT is inherently skewed to specific domains answerable for English-speaking area (*e.g.*, United States), which causes the domain gap with test data on a new language. Automatic labeler can complement by collecting unlabeled text on the domains not covered, and generating synthetic labels.

For generating synthetic training data, we leverage Question Generator (QG) and Answer Extractor (AE) as labeler, following (Du and Cardie, 2018). Given a passage in a target language, our goal is to generate question-answer pairs, related to the given passage. In (Du and Cardie, 2018), BiLSTM-CRF model (Huang et al., 2015) is used for AE, classifying whether the word belongs to the answer span. For QG, sequence-to-sequence model (Bahdanau et al., 2014) is adopted, by setting a passage and answer candidates as input and question words as output. To train QG and AE, we use weak labels obtained from Method I. At inference time of QG and AE, we insert human-written passages, crawled from web documents in a target language (*e.g.*, Wikipedia), and obtain pairs of question and answer as the output of QG and AE, respectively.

4 Weakly-supervised QA model

In this section, we illustrate how we score confidence for noisy training data obtained from the above two generation methods. We propose a new *Refinery* network, of not only estimating confi-

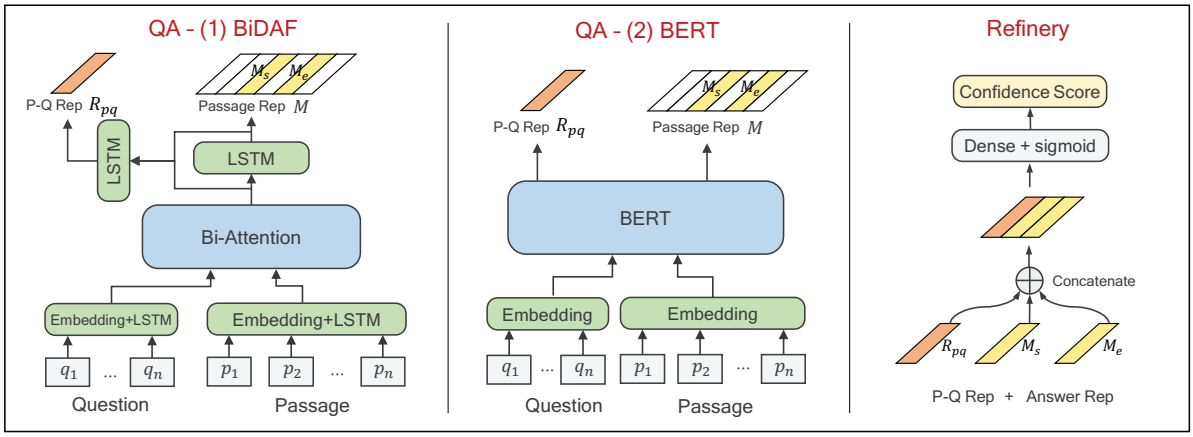


Figure 2: Architecture of our QA model and *Refinery*. Our approach of *Refinery* (Right) can be applicable to various RCQA models, illustrated with BiDAF (Left) and BERT (Center) here.

dence of noisy input from multiple generators but also improving labels and QA model. Existing work (Bach et al., 2011) estimates translation error that can be used as a confidence proxy. However, it requires large labels through human-correction, and cannot apply to other generators such as QG. Our goal is confidence estimation regardless of whether it is generated from NMT or QG methods (generator-agnostic).

4.1 Refinery Network

Our *Refinery* aims to score the quality of the generated (p, q, a) labels, depending on whether the example is positive or negative. It is known that RCQA models trained solely on positive examples fail on unanswerable cases, assuming that a correct answer is guaranteed to exist in the passage (Rajpurkar et al., 2018). A simple remedy is to manually augment training resources with negative examples, to teach how to distinguish them from positive examples. Our key contribution is automatic collection of both positive and negative training examples for training **confidence** prediction, which we will discuss in details below.

Our research question now is then: how do we obtain positive and negative examples? For positive examples, we treat weak labels from the generation methods as "pseudo-positive", since it would be positive, when the translation/generation is perfect. For negative examples, we generate synthetic noises, similarly to translation/generation errors, by adopting a naive but effective method to change words or sentences in a positive example (Levy et al., 2017; Zhang et al., 2019; Yang et al., 2019).

Specifically, we modify pseudo-positive labels to generate synthetic negative examples of unanswerable or wrong answer. For unanswerable cases, (1) we first replace a question with its semantic neighbor derived from other documents (*e.g.*, "who first discovered america?" into "who first discovered canada?"). For this, we use the average of word embedding and the cosine distance. Second, (2) we modify a passage by removing a sentence containing the answer span, so that the modified passage is the subset of original one, but unanswerable. Lastly, for wrong answers, (3) we perturb an answer span into a random span in the same passage, which is answerable but with an incorrect label. We generate negative examples from the above (1)-(3) methods (33.3% each). As we will show later in Section 5, these negative examples are empirically effective for training *Refinery*.

The training objective is that CS of pseudo-positive example should be greater than that of negative one, with at least margin δ. To deal with pseudo-positive examples unequally, we set the margin δ dynamically, which is derived from F1 score of QA model. That is, we relatively assign a larger margin to a higher confidence instance, and smaller otherwise. The magnitude of margin depends on the F1 score of the predicted answer from QA model. Considering the margin δ, the final objective function is computed as follows:

$$\delta_i = \alpha \cdot F1\ score(p_i^+, q_i^+, a_i^+)$$
$$L_{cs} = \sum_i max(0, \delta_i - \{CS_i^+ - CS_i^-\}) \quad (3)$$

where α is a hyper-parameter, CS_i^+ indicates the confidence of positive example, and CS_i^- is that of negative example.

Now, to obtain the confidence score, we use BiDAF (Seo et al., 2016) as QA model, as shown in Figure 2 (Left). First, to model answerability of (p, q), we extract the feature of (p, q) from QA model. For this, we concatenate the two hidden states M^1 and M^2 from BiDAF, as mentioned in Section 2.1. Then, the representations are aggregated by LSTM and attention layer, as follows:

$$\begin{aligned} G &= \sigma(W_1[M^1 ; M^2] + b_1) \\ \overline{G} &= \overleftrightarrow{LSTM}(G) \\ a &= softmax(v_1 \cdot \sigma(W_2\overline{G} + b_2)) \\ R_{pq} &= \sum_i a_i \cdot \overline{G_i} \end{aligned} \tag{4}$$

where W_1, W_2, v_1, b_1, b_2 are trainable weights, and σ is tanh function. The vector R_{pq} containing the information of both passage and question will be used for confidence score.

Second, to model the confidence of answer labels, we represent answer spans in the labels. For this, we concatenate two hidden states M_s^1 and M_e^2 of s-th start and e-th end words in the answer. The final confidence score CS is obtained from R_{pq} and the answer feature, as follows:

$$\begin{aligned} R_{span} &= [\ M_s^1 ; M_e^2\ ] \\ CS &= sigmoid(v_2 \cdot [\ R_{pq} ; R_{span}\ ] + b_3) \end{aligned} \tag{5}$$

where v_2 and b_3 are trainable weights, and $[\ ;\ ]$ indicates concatenation.

An extension to the above model is to replace BiDAF with BERT (Devlin et al., 2018), a pre-trained model on a masked language model task. Due to the commonality of BiDAF and BERT, taking question and passage as input and generating passage representations for answer span prediction, we can apply BERT to our approach, as shown in Figure 2 (Center), by modifying *Refinery* module of modeling confidence score. Meanwhile, architectural differences of the two, such as BiDAF using bi-attention layers and BERT using self-attention layers, require some minor changes: In BERT, as the output of CLS token can represent the input sequence of passage and question, we use the output vector of CLS as R_{pq}. For R_{span}, we use the hidden states at the layer of BERT, instead of M in Eq. (5). We will show the effectiveness of our approach on both BiDAF and BERT model in Section 5.

4.2 Improving QA and Weak Labels

During an iterative procedure of QA and *Refinery*, our approach improves both QA model and weak labels. Using confidence, *Refinery* acts as two functions: (1) **down-weighting** instances with low-quality, and (2) **answer modification** with low-quality for higher quality.

The objective of original QA model minimizes the sum of the negative log probabilities:

$$\begin{aligned} L_{qa} &= -\sum_i^N log(\mathbf{P}^1_{s_i}) + log(\mathbf{P}^2_{e_i}) \\ &= -\sum_i^N log(P(a_i|p_i, q_i)) \end{aligned} \tag{6}$$

where N is the number of examples in training set, s_i and e_i are the start and end indices of the i-th example, respectively. For down-weighting noise, we combine this QA loss function with weighted confidence score, based on function $f(CS(p, q))$:

$$L_{qa'} = -\sum_i^N f(CS_i) \cdot log(P(a_i|p_i, q_i)) \tag{7}$$

However, in initial steps of training, the confidence score is not reliable, and may cause unstable training. To avoid the problem, we apply an annealing technique, adjusting the contribution of the confidence score. We set the initial function f to be uniform constant, and then gradually increase the contribution of instance weighting as learning steps. That is, we design the function f as follows:

$$f(CS_i) = \frac{c \cdot e^{-\lambda t} + CS_i}{c \cdot e^{-\lambda t} + 1} \tag{8}$$

where t is the number of current iteration step, c and λ are hyper-parameters.

Besides down-weighting noisy instances, we could change some answers in weak labels to higher-quality one, while unanswerable passages-question pairs are difficult to modify. To improve such labels, we compare confidence scores between the answer a^* predicted by QA model and a in the weak label. If the confidence increase is greater than threshold γ, we change the answer a to the model-generated answer a^* when each epoch ends. Finally, we train the QA and *Refinery* modules jointly, as Eq. (3) and (7), which are updated in turns[3], until convergence. Algorithm 1 describes this procedure in detail.

[3]When updating *Refinery*, we freeze parameters in QA model, because negative examples could be harmful in QA

Algorithm 1 Iterative QA and Refinery

$N = 0$, $\theta_{qa}^{(0)} \leftarrow$ Initial QA, $\theta_{re}^{(0)} \leftarrow$ Initial Refinery
for *epoch* ***in*** *epochs* **do**
  for i ***in*** *max steps* **do**
    Let D be (P, Q, A) triplets in mini-batch
    $D_i^+ \leftarrow (P_i^+, Q_i^+, A_i^+)$ from weak labels
    $D_i^- \leftarrow (P_i^-, Q_i^-, A_i^-)$ from negative examples
    $A_i^* \leftarrow$ Answer of (P_i^+, Q_i^+) from QA $\theta_{qa}^{(N)}$
    $F_i \leftarrow$ F1 score of (A_i^+, A_i^*)
    $CS_i \leftarrow$ Confidence(D_i^+) from Refinery $\theta_{re}^{(N)}$
    $\theta_{qa}^{(N+1)} \leftarrow$ Update QA model on (D_i^+, CS_i)
    $\theta_{re}^{(N+1)} \leftarrow$ Update Refinery on (D_i^+, D_i^-, F_i)
    $N \leftarrow N + 1$
  end
  $D^* \leftarrow (P^+, Q^+, A^*)$ predicted from QA $\theta_{qa}^{(N)}$
  $CS^* \leftarrow$ Confidence(D^*) on predicted answers
  $CS^+ \leftarrow$ Confidence(D^+) on weak labels
  if $CS^* - CS^+ > \gamma$ **then**
    $D^+ \leftarrow$ modified label (P^+, Q^+, A^*)
  end
end

5 Experiment

In this section, we address the following research questions:

- RQ1: Does our proposed work outperform existing approaches? Does it generalize for languages with diverse distance or topics?
- RQ2: Does our model generally work on more noisy environment?
- RQ3: Is our *Refinery* effective in distinguishing positive and negative set? How does QA/*Refinery* contribute to each other?

5.1 Dataset

For evaluation, we conduct experiments on three datasets for French and Korean RCQA. First is the public datasets, built on French (327 pairs) (Asai et al., 2018) and Korean (2K pairs) (Lee et al., 2018) Wikipedia, denoted as **Wiki_Fr** and **Wiki_Kr** respectively. Supposing zero-annotation at training, we use their training set (2K) in **Wiki_Kr** as our dev set, and evaluate our model on the test set. Third dataset is a history RC dataset (7K), which we collect on Korean-specific topics to show the robustness of our model with respect to domain gap: Specifically, dealing with Korean history documents, as denoted as **History_Kr**. For annotation, we follow the convention of SQuAD (Rajpurkar et al., 2016). Lastly, for generating synthetic labels, we crawled about 400 articles in each French and Korean, with general and (Korean) history topic from Wikipedia and Doopedia[4], respectively. Through automatic labeler (QG/AE) as mentioned in Section 3.2, we generated totally 100K QA pairs from passages in Wikipedia and Doopedia.

5.2 Experimental Setting

Our implementation settings for QA model follow original BiDAF (Seo et al., 2016) and BERT (Multilingual-Base version) (Devlin et al., 2018). For machine translation, we use pre-trained and open-sourced NMT models for French[5] and Korean[6]. As hyper-parameters, α, c, λ and γ, are set to 1/10, 10, 1/4000, and 0.5 respectively, optimized by dev set. For QG and AE implementations, we follow the setting of NQG model (Zhou et al., 2017). In BiDAF model, we used Natural Language Toolkit (NLTK) and KoNLPY[7] for French/Korean tokenizer, and FastText (Mikolov et al., 2017) for word embedding.

5.3 Evaluation of Full QA Model

We compare our QA model with the following baselines:

Base1&2: Asai et al. (2018) translate (p, q, a) in target language into English, then test on pre-trained English QA model. The answer from model is translated back to target language. We present their reported F1 scores for French with and without ELMo (Peters et al., 2018), and leave unreported results blank.

Base3: Lee et al. (2018) proposed semi-supervised method, to train the weak model on a small human-labeled set, and evaluate translated QA pairs by the output probabilities (in Eq. (2)) of the weak QA model. For training QA, the translated data over threshold is used together with a human-labeled data. We present their reported F1 scores and leave unreported results blank.

Ours: Our proposed models are trained on BiDAF and BERT, and with/without Refinery (for ablation purposes described below).

[4] https://www.doopedia.co.kr
[5] https://github.com/pytorch/fairseq/blob/master/examples/translation
[6] https://developers.naver.com/products/nmt/
[7] https://konlpy-ko.readthedocs.io/

Table 1: The results of QA models. Compared to version w/o Refinery, our models outperform with statistical significance (* indicates $p < 0.05$).

	Method	QA model	F1 score		
			Wiki_Fr	Wiki_Kr	History_Kr
Base1	Test on source (Asai et al., 2018)	BiDAF+ELMo	61.9	-	-
Base2	Test on source (Asai et al., 2018)	BiDAF	57.6	-	-
Base3	Semi-supervision (Lee et al., 2018)	BiDAF	-	71.5	-
Ours	No weighting (w/o *Refinery*)	BiDAF	60.8	70.1	61.5
	Full model with *Refinery*	BiDAF	63.5*	73.4*	64.1*
	No weighting (w/o *Refinery*)	BERT	74.3	77.1	67.7
	Full model with *Refinery*	BERT	**76.6***	**79.7***	**70.4***

Results on QA datasets Table 1 shows the comparison on En-Kr (distant) and En-Fr (close) language pairs. Similarly, we contrast in diverse topic-similarity scenarios: History (distant) and Wiki (close). When compared with Base1&2 (Test on source), our model outperforms the two models, even when Base1 is boosted by ELMo (Peters et al., 2018). When compared with Base3, semi-supervision (Lee et al., 2018) using 2K human labels outperforms our BiDAF model (without *Refinery*), on Wiki_Kr set. However, ours with *Refinery* outperforms the Base3 model, which means that large and noisy labels refined by our method have comparable quality to small but strong labels.

To show that the effectiveness of our approach is orthogonal to QA model, we construct two QA models, BiDAF and BERT, as in Table 1. When compared with *No weighting*, our full model on BiDAF improved 2.7%, 3.3% and 2.6% of F1 scores, and that on BERT improved 2.3%, 2.6% and 2.7% of F1 score, on Wiki_Fr, Wiki_Kr and History_Kr, respectively. BERT model with our *Refinery* performs the best among all models, by adding the power of pre-trained representations.

Ablation Study As shown in Table 2, we conduct an ablation study on both BiDAF and BERT models, by examining the effects, from removing each component. In (A), we replace our confidence score (CS) with the probabilities in Eq. (2), which is similar to the use of confidence in (Lee et al., 2018). We normalize the probabilities in mini-batch by softmax function, then apply to *down-weighting*. On the QA model using the replaced confidence, the performance decreased significantly, suggesting the importance of CS. In (B), we remove a dynamic margin in Eq. (3), by setting the margin as a constant value instead (*i.e.*, $\delta = 1$). Using static margin lowered the performance of both QA models, suggesting that QA feedback (dynamic margin) is effective for refining noises. In (C), we remove *answer modification* in Section 4.2, while preserving *down-weighting* module. Compared with our full model, adding *answer modification* improved the performance on both QA models. We can also observe the effect of *down-weighting*, as the model without *answer modification* still outperforms that with *No weighting* in Table 1.

Table 2: The ablation study on three datasets. The number inside the parenthesis indicates the decrease from our full model.

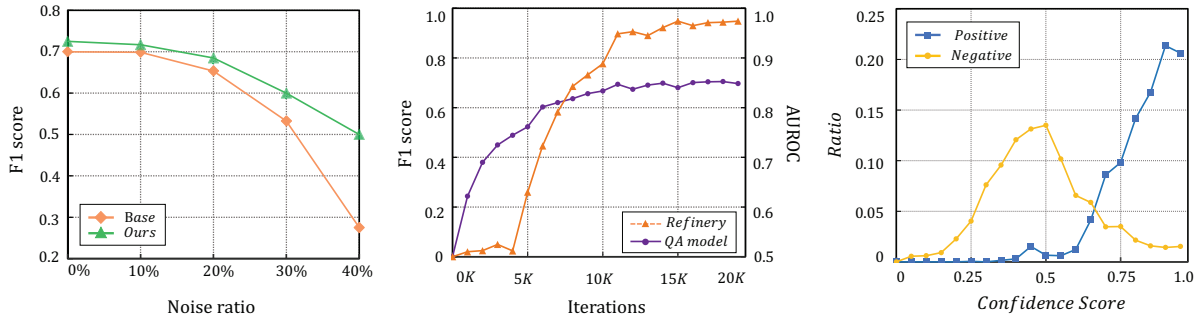
(a) On BiDAF

		F1 score		
		Wiki_Fr	Wiki_Kr	History_Kr
	Our full model	63.5	73.4	64.1
(A)	Replace CS with Prob	61.1 (-2.4)	70.3 (-3.1)	61.8 (-2.3)
(B)	Remove dynamic margin	62.0 (-1.5)	72.3 (-1.1)	62.4 (-1.7)
(C)	Remove answer modification	62.3 (-1.2)	71.9 (-1.5)	62.8 (-1.3)

(b) On BERT

		F1 score		
		Wiki_Fr	Wiki_Kr	History_Kr
	Our full model	76.6	79.7	70.4
(A)	Replace CS with Prob	74.3 (-2.3)	77.3 (-2.4)	67.8 (-2.6)
(B)	Remove dynamic margin	75.1 (-1.5)	77.8 (-1.9)	68.4 (-2.0)
(C)	Remove answer modification	75.2 (-1.4)	78.6 (-1.1)	69.2 (-1.2)

For RQ2, we design scenarios with noisier training data to demonstrate the robustness. When training QA, we replace positive examples with negative examples, perturbing an input to fool a machine model. Figure 3(a) shows the robustness of our BiDAF model over varying noise ratios. As



(a) QA performance as noise ratio (b) Iterative QA and *Refinery* (c) Distribution of confidence score

Figure 3: Experimental results of our model on dev set. (a) Comparison of robustness between Base (w/o *Refinery*) and Ours in noisy environments. (b) The performances of QA (F1 score) and *Refinery* (AUROC) over iterations. (c) Distribution of confidence scores for positive and negative examples.

Table 3: The results of confidence score evaluation

	TNR (95% TPR)	AUROC	AUPR
	(Wiki_Kr / History_Kr)		
(A) Base1	47.4 / 39.0	82.9 / 85.8	78.2 / 85.4
(B) Base2	44.9 / 69.5	71.2 / 83.5	63.3 / 75.7
Ours	**88.4 / 90.9**	**97.7 / 98.5**	**98.2 / 98.8**

noises increase, Base BiDAF model sharply drops its F1 score, while that of our full model degrades gracefully.

5.4 Evaluation of Confidence Score

This section addresses RQ3, aiming at directly observing whether our confidence score effectively distinguishes the positive from negative. As baselines, we conduct (A) and (B) in the above ablation study. (A) Base1 uses the probability in Eq. (2) as confidence score. (B) Base2 is *Refinery* without a dynamic margin ($\delta = 1$). In this experiment, we use BiDAF model on Wiki_Kr and History_Kr, and generate negative examples from positive test set as 1:1 ratio.

As a quantitative evaluation, we measure three metrics of detecting negative examples, comparing confidence of positive and negative one.

- True Negative Rate (TNR) at 95% True Positive Rate (TPR).
- Area under the Receiver Operating Characteristic Curve (AUROC).
- Area under the Precision – Recall Curve (AUPR).

As shown in Table 3, our *Refinery* outperforms other baselines with statistical significance of $p <$ 0.01 and by a large margin on all measures, showing *Refinery* is effective in distinguishing positive-negative pairs. We also compare our work with and without a dynamic margin from QA feedback, to show that such strategy improves the performance.

To show performance of QA/*Refinery* over iterations, we check their performance as iterations continue on dev set, in Figure 3(b). In early iterations, *Refinery* cannot contribute to QA, because F1 score of all instance is zero. After the training stabilizes, the performance of *Refinery* increases rapidly. Figure 3(c) contrasts the distribution of confidence from positive and negative samples: Unlike negative examples of varying confidence, positive examples are clearly skewed to high-ends.

Table 4 compares the examples in Baseline and our model, where the first two are high-quality data and the third and forth are not. In the first example, both models show high score, while Baseline underestimates the quality and *Refinery* works correctly for the second example. The third example is unanswerable on (p, q) pair, since the words about the subject disappeared during translation, which is assigned to lower scores by two models. The last example has also translation error where the given question was mistranslated from 'when' to 'how'. In this case, our *Refinery* network successfully assigned low confidence score, as the question is no longer answerable after a bad translation, but Baseline gives a high confidence.

6 Related Work

Multilingual Task: NMT has played an important role in addressing multilinguality. For example, a straightforward solution for RCQA is translating (p, q) in target language to English and ap-

Table 4: Qualitative examples showing our proposed confidence estimation score CS for (p,q) pairs in Korean. Human translated English is given in the bracket only for readability. We omitted some irrelevant sentences in the passages for conciseness.

	Passage [**Answer**] / Question	Baseline $P(a\|p,q)$	CS
P1	Kor: 피셔와 그의 팀 멤버들은 **[요서프의 꿈]**으로 돌아오는데 성공했다. {...} Eng: Fischer and his team members are successful in returning to **[Yusuf's dream]**. {...}	0.877	0.978
Q1	Kor: 피셔와 그의 팀원들은 어디로 돌아왔나요? Eng: Where did Fischer and his team return?		
P2	Kor: 에릭은 **[헨리 앤 사우스 옥스퍼드셔 스탠다드]** 에 두 편의 시를 기고하였다.{...} Eng: Eric published two poems in **[the Henley and South Oxfordshire Standard]**. {...}	0.010	0.981
Q2	Kor: 에릭은 어디에 시를 기고했나요? Eng: Where did Eric publish his poems?		
P3	Kor: 그러나 공대는 **[1920년]**에 설립되었다. {...} Eng: But, the engineering college was founded in **[1920]**. {...}	0.015	0.053
Q3	Kor: 노트르담 대학은 몇 년도에 설립되었는가? Eng: When was University of Notre Dame founded?		
P4	Kor: **[2016년 2월 6일]** 비욘세는 음악 스트리밍 서비스만을 위해 새로운 싱글을 발매했다. {...} Eng: **[On February 6, 2016]**, Beyonce released a new single only for a music streaming.	0.864	0.090
Q4	Kor: 싱글 앨범은 어떻게 발매 되었나요? Eng: How was the single released?		

ply English-based models, then translate the answer back (Asai et al., 2018). Not only for question answering, such method has been successful in sentiment classication (Zhou et al., 2016), relation extraction (Faruqui and Kumar, 2015), and causal commonsense (Yeo et al., 2018). However, these approaches are dependent on quality of translation.

RCQA for Resource-Poor Language To overcome the lack of RCQA resources on other language, in (Lee et al., 2018), they proposed a semi-supervised strategy to remove the noise instances. They hand-annotated a small seed set of QA pairs to train a weak QA. Then, the weak QA indirectly judges the quality of machine-translated SQuAD pairs, by considering the output probability as confidence score. As confidence score, they selectively remove translated pairs below a fixed threshold. Meanwhile, we eliminate the needs for human annotation, by leveraging positive and negative sets from translated and generated data.

Combining models of QG and QA: Several works (Wang et al., 2017; Tang et al., 2017; Tang et al.) propose a framework jointly optimizing QG and QA model as dual. These approaches share commonality with our approach for generating training data with another model, but with the following crucial difference: Existing work can use a reliable training set annotated by human, which can be strong hint to decide the quality of generated data. For example, in (Tang et al.), QG generates questions on the given answer sentence, then the generated question is compared with the original question in training set. Meanwhile, we do not require human annotation, and rather leverage automatically generated from NMT and QG models, judiciously guided by our Refinery network.

7 Conclusion

This paper studies a zero-resource training data generation for supporting RCQA to a new target language. Inspired by limited resources, we generate RCQA data, by using existing NMT and QG techniques. To explore the noisy generated data, we proposed an integrated QA model with *Refinery* to control the generated data as confidence score. Our results showed that our strategies using *Refinery* and generated data enhance the performance of existing QA model, and *Refinery* has effectiveness to distinguish positive-negative pairs.

Acknowledgements

We thank NAVER Corporation for sponsoring this project. Partial support received from IITP grant funded by the Korea government (MSIT) (No.2017-0-01779, XAI).

References

Akari Asai, Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2018. Multilingual extractive

reading comprehension by runtime machine translation. *arXiv preprint arXiv:1809.03275*.

Nguyen Bach, Fei Huang, and Yaser Al-Onaizan. 2011. Goodness: A method for measuring machine translation confidence. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 211–219. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

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 and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from wikipedia. *arXiv preprint arXiv:1805.05942*.

Manaal Faruqui and Shankar Kumar. 2015. Multilingual open relation extraction using cross-lingual projection. *arXiv preprint arXiv:1503.06450*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Seung-won Hwang and Kevin Chang. 2005. Optimizing access cost for top-k queries over web sources: A unified cost-based approach. In *ICDE*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gáabor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association of Computational Linguistics*, 6:317–328.

Kyungjae Lee, Kyoungho Yoon, Sunghyun Park, and Seung-won Hwang. 2018. Semi-supervised training data generation for multilingual question answering. *LREC*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2018. Korquad: Korean qa dataset for machine comprehension. *Communications of the Korean Institute of Information Scientists and Engineers*, pages 539–541.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2018. Coqa: A conversational question answering challenge. *arXiv preprint arXiv:1808.07042*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 588–598.

Duyu Tang, Nan Duan, Tao Qin, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint arXiv:1706.02027*.

Duyu Tang, Nan Duan, Zhao Yan, Zhirui Zhang, Yibo Sun, Shujie Liu, Yuanhua Lv, and Ming Zhou. Learning to collaborate for question answering and asking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Ferhan Ture and Elizabeth Boschee. 2016. Learning to translate for multilingual question answering. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 573–584.

Tong Wang, Xingdi Yuan, and Adam Trischler. 2017. A joint model for question answering and question generation. *arXiv preprint arXiv:1706.01450*.

Wei Yang, Yuqing Xie, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. Data augmentation for bert fine-tuning in open-domain question answering. *arXiv preprint arXiv:1904.06652*.

Jinyoung Yeo, Geungyu Wang, Hyunsouk Cho, Seungtaek Choi, and Seung-won Hwang. 2018. Machine-translated knowledge transfer for commonsense causal reasoning. In *AAAI*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*.

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.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Attention-based lstm network for cross-lingual sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 247–256.