Incorporating External Knowledge into Machine Reading for Generative Question Answering

Bin Bi, Chen Wu, Ming Yan, Wei Wang,
Jiangnan Xia, Chenliang Li
Alibaba Group
{b.bi, wuchen.wc, ym119608, hebian.ww}@alibaba-inc.com
{jiangnan.xjn, lcl193798}@alibaba-inc.com

Abstract

Commonsense and background knowledge is required for a QA model to answer many nontrivial questions. Different from existing work on knowledge-aware QA, we focus on a more challenging task of leveraging external knowledge to generate answers in natural language for a given question with context.

In this paper, we propose a new neural model, Knowledge-Enriched Answer Generator (KEAG), which is able to compose a natural answer by exploiting and aggregating evidence from all four information sources available: question, passage, vocabulary and knowledge. During the process of answer generation, KEAG adaptively determines when to utilize symbolic knowledge and which fact from the knowledge is useful. This allows the model to exploit external knowledge that is not explicitly stated in the given text, but that is relevant for generating an answer. The empirical study on public benchmark of answer generation demonstrates that KEAG improves answer quality over models without knowledge and existing knowledge-aware models, confirming its effectiveness in leveraging knowledge.

1 Introduction

Question Answering (QA) has come a long way from answer sentence selection, relational QA to machine reading comprehension. The next-generation QA systems can be envisioned as the ones which can read passages and write long and abstractive answers to questions. Different from extractive question answering, generative QA based on machine reading produces an answer in true natural language which does not have to be a sub-span in the given passage.

Most existing models, however, answer questions based on the content of given passages as the only information source. As a result, they may not be able to understand certain passages or to answer certain questions, due to the lack of commonsense and background knowledge, such as the knowledge about what concepts are expressed by the words being read (lexical knowledge), and what relations hold between these concepts (relational knowledge). As a simple illustration, given the passage:

State officials in Hawaii on Monday said they have once again checked and confirmed that President Barack Obama was born in Hawaii.

to answer the question: *Was Barack Obama born in the U.S.?*, one must know (among other things) that Hawaii is a state in the U.S., which is external knowledge not present in the text corpus.

Therefore, a QA model needs to be enriched with external knowledge properly to be able to answer many nontrivial questions. Such knowledge can be commonsense knowledge or factual background knowledge about entities and events that is not explicitly expressed but can be found in a knowledge base such as ConceptNet (Speer et al., 2016), Freebase (Pellissier Tanon et al., 2016) and domain-specific KBs collected by information extraction (Fader et al., 2011; Mausam et al., 2012). Thus, we aim to design a neural model that encodes pre-selected knowledge relevant to given questions, and that learns to include the available knowledge as an enrichment to given textual information.

In this paper, we propose a new neural architecture, Knowledge-Enriched Answer Generator (KEAG), specifically designed to generate natural answers with integration of external knowledge. KEAG is capable of leveraging symbolic knowledge from a knowledge base as it generates each word in an answer. In particular, we assume that each word is generated from one of the four information sources: 1. question, 2. passage, 3. vocabulary and 4. knowledge. Thus, we introduce the

source selector, a sentinel component in KEAG that allows flexibility in deciding which source to look to generate every answer word. This is crucial, since knowledge plays a role in certain parts of an answer, while in others text context should override the context-independent knowledge available in general KBs.

At each timestep, before generating an answer word, KEAG determines an information source. If the knowledge source is selected, the model extracts a set of facts that are potentially related to the given question and context. A stochastic fact selector with discrete latent variables then picks a fact based on its semantic relevance to the answer being generated. This enables KEAG to bring external knowledge into answer generation, and to generate words not present in the predefined vocabulary. By incorporating knowledge explicitly, KEAG can also provide evidence about the external knowledge used in the process of answer generation.

We introduce a new differentiable sampling-based method to learn the KEAG model in the presence of discrete latent variables. For empirical evaluation, we conduct experiments on the benchmark dataset of answer generation MARCO (Nguyen et al., 2016). The experimental results demonstrate that KEAG effectively leverages external knowledge from knowledge bases in generating natural answers. It achieves significant improvement over classic QA models that disregard knowledge, resulting in higher-quality answers.

2 Related Work

There have been several attempts at using machine reading to generate natural answers in the QA field. Tan et al. (2018) took a generative approach where they added a decoder on top of their extractive model to leverage the extracted evidence for answer synthesis. However, this model still relies heavily on the extraction to perform the generation and thus needs to have start and end labels (a span) for every QA pair. Mitra (2017) proposed a seq2seq-based model that learns alignment between a question and passage words to produce rich question-aware passage representation by which it directly decodes an answer. Gao et al. (2019) focused on product-aware answer generation based on large-scale unlabeled e-commerce reviews and product attributes. Furthermore, natural answer generation can be refor-

mulated as query-focused summarization which is addressed by Nema et al. (2017).

The role of knowledge in certain types of QA tasks has been remarked on. Mihaylov and Frank (2018) showed improvements on a cloze-style task by incorporating commonsense knowledge via a context-to-commonsense attention. Zhong et al. (2018) proposed commonsense-based pre-training to improve answer selection. Long et al. (2017) made use of knowledge in the form of entity descriptions to predict missing entities in a given document. There have also been a few studies on incorporating knowledge into QA models without passage reading. GenQA (Yin et al., 2016) combines knowledge retrieval and seq2seq learning to produce fluent answers, but it only deals with simple questions containing one single fact. COREQA (He et al., 2017) extends it with a copy mechanism to learn to copy words from a given question. Moreover, Fu and Feng (2018) introduced a new attention mechanism that attends across the generated history and memory to explicitly avoid repetition, and incorporated knowledge to enrich generated answers.

Some work on knowledge-enhanced natural language (NLU) understanding can be adapted to the question answering task. CRWE (Weissenborn, 2017) dynamically integrates background knowledge in a NLU model in the form of free-text statements, and yields refined word representations to a task-specific NLU architecture that reprocesses the task inputs with these representations. In contrast, KBLSTM (Yang and Mitchell, 2017) leverages continuous representations of knowledge bases to enhance the learning of recurrent neural networks for machine reading. Furthermore, Bauer et al. (2018) proposed MHPGM, a QA architecture that fills in the gaps of inference with commonsense knowledge. The model, however, does not allow an answer word to come directly from knowledge. We adapt these knowledge-enhanced NLU architectures to answer generation, as baselines for our experiments.

3 Knowledge-aware Answer Generation

Knowledge-aware answer generation is a question answering paradigm, where a QA model is expected to generate an abstractive answer to a given question by leveraging both the contextual passage and external knowledge. More formally, given a knowledge base $\mathcal{K}$ and two sequences of input words: question
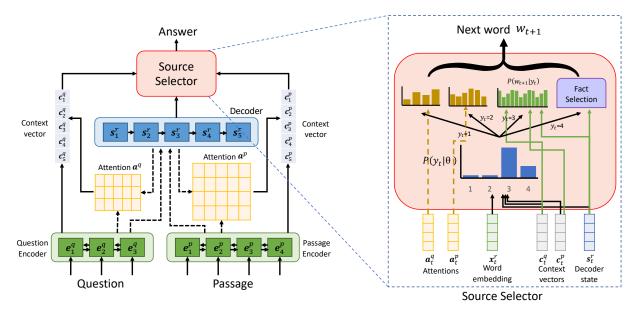
Figure 1: An overview of the architecture of KEAG (best viewed in color). A question and a passage both go through an extension of the sequence-to-sequence model. The outcomes are then fed into a source selector to generate a natural answer.

$q = \{w_1^q, w_2^q, \ldots, w_{N_q}^q\}$ and passage $p = \{w_1^p, w_2^p, \ldots, w_{N_p}^p\}$, the answer generation model should produce a series of answer words $r = \{w_1^r, w_2^r, \ldots, w_{N_r}^r\}$. The knowledge base $\mathcal{K}$ contains a set of facts, each of which is represented as a triple $f = (subject, relation, object)$ where $subject$ and $object$ can be multi-word expressions and $relation$ is a relation type, e.g., $(bridge, UsedFor, cross\ water)$.

3.1 Knowledge-Enriched Answer Generator

To address the answer generation problem, we propose a novel KEAG model which is able to compose a natural answer by recurrently selecting words at the decoding stage. Each of the words comes from one of the four sources: question q, passage p, global vocabulary $\mathcal{V}$, and knowledge $\mathcal{K}$. In particular, at every generation step, KEAG first determines which of the four sources to inspect based on the current state, and then generates a new word from the chosen source to make up a final answer. An overview of the neural architecture of KEAG is depicted in Figure 1.

3.2 Sequence-to-sequence model

KEAG is built upon an extension of the sequence-to-sequence attentional model (Bahdanau et al., 2015; Nallapati et al., 2016; See et al., 2017). The words of question q and passage p are fed one-by-one into two different encoders, respectively. Each of the two encoders, which are both bidirectional LSTMs, produces a sequence of encoder hidden

states ($\mathbf{E}^q$ for question q, and $\mathbf{E}^p$ for passage p). In each timestep t, the decoder, which is a unidirectional LSTM, takes an answer word as input, and outputs a decoder hidden state $\mathbf{s}_t^r$.

We calculate attention distributions $\mathbf{a}_t^q$ and $\mathbf{a}_t^p$ on the question and the passage, respectively, as in (Bahdanau et al., 2015):

$$\mathbf{a}_t^q = \text{softmax}(\mathbf{g}^{q\mathsf{T}}\tanh(\mathbf{W}^q\mathbf{E}^q + \mathbf{U}^q\mathbf{s}_t^r + \mathbf{b}^q)), \tag{1}$$

$$\mathbf{a}_t^p = \text{softmax}(\\ \mathbf{g}^{p\mathsf{T}}\tanh(\mathbf{W}^p\mathbf{E}^p + \mathbf{U}^p\mathbf{s}_t^r + \mathbf{V}^p\mathbf{c}^q + \mathbf{b}^p)), \tag{2}$$

where $\mathbf{g}^q$, $\mathbf{W}^q$, $\mathbf{U}^q$, $\mathbf{b}^q$, $\mathbf{g}^p$, $\mathbf{W}^p$, $\mathbf{U}^p$ and $\mathbf{b}^p$ are learnable parameters. The attention distributions can be viewed as probability distributions over source words, which tells the decoder where to look to generate the next word. The coverage mechanism is added to the attentions to avoid generating repetitive text (See et al., 2017). In Equation 2, we introduce $\mathbf{c}^q$, a context vector for the question, to make the passage attention aware of the question context. $\mathbf{c}^q$ for the question and $\mathbf{c}^p$ for the passage are calculated as follows:

$$\mathbf{c}_t^q = \sum_i a_{ti}^q \cdot \mathbf{e}_i^q, \qquad \mathbf{c}_t^p = \sum_i a_{ti}^p \cdot \mathbf{e}_i^p, \tag{3}$$

where $\mathbf{e}_i^q$ and $\mathbf{e}_i^p$ are an encoder hidden state for question q and passage p, respectively. The context vectors ($\mathbf{c}_t^q$ and $\mathbf{c}_t^p$) together with the attention distributions ($\mathbf{a}_t^q$ and $\mathbf{a}_t^p$) and the decoder state

$(\mathbf{s}_t^r)$ will be used downstream to determine the next word in composing a final answer.

4 Source Selector

During the process of answer generation, in each timestep, KEAG starts with running a source selector to pick a word from one source of the question, the passage, the vocabulary and the knowledge. The right plate in Figure 1 illustrates how the source selector works in one timestep during decoding.

If the question source is selected in timestep t, KEAG picks a word according to the attention distribution $\mathbf{a}_t^q \in \mathbb{R}^{N_q}$ over question words (Equation 1), where N_q denotes the number of distinct words in the question. Similarly, when the passage source is selected, the model picks a word from the attention distribution $\mathbf{a}_t^p \in \mathbb{R}^{N_p}$ over passage words (Equation 2), where N_p denotes the number of distinct words in the passage. If the vocabulary is the source selected in timestep t, the new word comes from the conditional vocabulary distribution $P_v(w|\mathbf{c}_t^q, \mathbf{c}_t^p, \mathbf{s}_t^r)$ over all words in the vocabulary, which is obtained by:

$$P_v(w|\mathbf{c}_t^q, \mathbf{c}_t^p, \mathbf{s}_t^r) = \text{softmax}(\mathbf{W}^v \cdot [\mathbf{c}_t^q, \mathbf{c}_t^p, \mathbf{s}_t^r] + \mathbf{b}^v), \tag{4}$$

where $\mathbf{c}_t^q$ and $\mathbf{c}_t^p$ are context vectors, and $\mathbf{s}_t^r$ is a decoder state. $\mathbf{W}^v$ and $\mathbf{b}^v$ are learnable parameters.

To determine which of the four sources a new word w_{t+1} is selected from, we introduce a discrete latent variable $y_t \in \{1, 2, 3, 4\}$ as an indicator. When $y_t = 1$ or 2, the word w_{t+1} is generated from the distribution $P(w_{t+1}|y_t)$ given by:

$$P(w_{t+1}|y_t) = \begin{cases} \sum_{i:w_i=w_{t+1}} a_{ti}^q & y_t = 1 \\ \sum_{i:w_i=w_{t+1}} a_{ti}^p & y_t = 2. \end{cases} \tag{5}$$

If $y_t = 3$, KEAG picks word w_{t+1} according to the vocabulary distribution $P_v(w|\mathbf{c}_t^q, \mathbf{c}_t^p, \mathbf{s}_t^r)$ given in Equation 4. Otherwise, if $y_t = 4$, the word w_{t+1} comes from the fact selector, which will be described in the coming section.

5 Knowledge Integration

In order for KEAG to integrate external knowledge, we first extract related facts from the knowledge base in response to a given question, from which we then pick the most relevant fact that can be used for answer composition. In this section, we present the two modules for knowledge integration: *related fact extraction* and *fact selection*.

5.1 Related Fact Extraction

Due to the size of a knowledge base and the large amount of unnecessary information, we need an effective way of extracting a set of candidate facts which provide novel information while being related to a given question and passage.

For each instance (q, p), we first extract facts with the subject or object that occurs in question q or passage p. Scores are added to each extracted fact according to the following rules:

- Score+4, if the subject occurs in q, *and* the object occurs in p.
- Score+2, if the subject *and* the object both occur in p.
- Score+1, if the subject occurs in q *or* p.

The scoring rules are set heuristically such that they model relative fact importance in different interactions. Next, we sort the fact triples in descending order of their scores, and take the top N_f facts from the sorted list as the related facts for subsequent processing.

5.2 Fact Selection

Figure 2 displays how a fact is selected from the set of related facts for answer completion. With the extracted knowledge, we first embed every related fact f by concatenating the embeddings of the subject $\mathbf{e}^s$, the relation $\mathbf{e}^r$ and the object $\mathbf{e}^o$. The embeddings of subjects and objects are initialized with pre-trained GloVe vectors (and average pooling for multiple words), when the words are present in the vocabulary. The fact embedding is followed by a linear transformation to relate subject $\mathbf{e}^s$ to object $\mathbf{e}^o$ with relation $\mathbf{e}^r$:

$$\mathbf{f} = \mathbf{W}^e \cdot [\mathbf{e}^s, \mathbf{e}^r, \mathbf{e}^o] + \mathbf{b}^e. \tag{6}$$

where $\mathbf{f}$ denotes fact representation, $[\cdot, \cdot]$ denotes vector concatenation, and $\mathbf{W}^e$ and $\mathbf{b}^e$ are learnable parameters. The set of all related fact representations $F = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{N_f}\}$ is considered to be a short-term memory of the knowledge base while answering questions on given passages.

To enrich KEAG with the facts collected from the knowledge base, we propose to complete an answer with the most relevant fact(s) whenever it is determined to resort to knowledge during the process of answer generation. The most relevant fact is selected from the related fact set F based on the dynamic generation state. In this model, we introduce a discrete latent random variable $z_t \in$
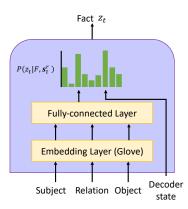
Figure 2: An overview of the fact selection module (best viewed in color)

$[1, N_f]$ to explicitly indicate which fact is selected to be put into an answer in timestep t. The model selects a fact by sampling a z_t from the discrete distribution $P(z_t|F, \mathbf{s}_t^r)$ given by:

$$P(z_t|\cdot) = \frac{1}{Z} \cdot \exp(\mathbf{g}^{f\intercal}\tanh(\mathbf{W}^f\mathbf{f}_{z_t} + \mathbf{U}^f\mathbf{s}_t^r + \mathbf{b}^f)), \tag{7}$$

where Z is the normalization term, $Z = \sum_{i=1}^{N_f} \exp(\mathbf{g}^{f\intercal}\tanh(\mathbf{W}^f\mathbf{f}_i + \mathbf{U}^f\mathbf{s}_t^r + \mathbf{b}^f))$, and $\mathbf{s}_t^r$ is the hidden state from the decoder in timestep t. $\mathbf{g}^f$, $\mathbf{W}^f$, $\mathbf{U}^f$ and $\mathbf{b}^f$ are learnable parameters.

The presence of discrete latent variables $\mathbf{z}$, however, presents a challenge to training the neural KEAG model, since the backpropagation algorithm, while enabling efficient computation of parameter gradients, does not apply to the non-differentiable layer introduced by the discrete variables. In particular, gradients cannot propagate through discrete samples from the categorical distribution $P(z_t|F, \mathbf{s}_t^r)$.

To address this problem, we create a differentiable estimator for discrete random variables with the Gumbel-Softmax trick (Jang et al., 2017). Specifically, we first compute the discrete distribution $P(z_t|F, \mathbf{s}_t^r)$ with class probabilities $\pi_1, \pi_2, \ldots, \pi_{N_f}$ by Equation 7. The Gumbel-Max trick (Gumbel, 1954) allows to draw samples from the categorical distribution $P(z_t|F, \mathbf{s}_t^r)$ by calculating one_hot$(\arg\max_i[g_i + \log\pi_i])$, where $g_1, g_2, \ldots, g_{N_f}$ are i.i.d. samples drawn from the Gumbel$(0, 1)$ distribution. For the inference of a discrete variable z_t, we approximate the Gumbel-Max trick by the continuous softmax function (in place of $\arg\max$) with temperature τ to generate a sample vector $\hat{\mathbf{z}}_t$:

$$\hat{z}_{ti} = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^{N_f} \exp((\log(\pi_j) + g_j)/\tau)}. \tag{8}$$

When τ approaches zero, the generated sample $\hat{\mathbf{z}}_t$ becomes a one-hot vector. τ is gradually annealed over the course of training.

This new differentiable estimator allows us to backpropagate through $z_t \sim P(z_t|F, \mathbf{s}_t^r)$ for gradient estimation of every single sample. The value of z_t indicates a fact selected by the decoder in timestep t. When the next word is determined to come from knowledge, the model appends the object of the selected fact to the end of the answer being generated.

6 Learning Model Parameters

To learn the parameters θ in KEAG with latent source indicators $\mathbf{y}$, we maximize the log-likelihood of words in all answers. For each answer, the log-likelihood of the words is given by:

$$\log P(w_1^r, w_2^r, \ldots, w_{N_r}^r|\theta) = \sum_{t=1}^{N_r} \log P(w_t^r|\theta)$$

$$= \sum_{t=1}^{N_r} \log \sum_{y_t=1}^{4} P(w_{t+1}|y_t)P(y_t|\theta) \tag{9}$$

$$\geq \sum_{t=1}^{N_r} \sum_{y_t=1}^{4} P(y_t|\theta) \log P(w_{t+1}|y_t) \tag{10}$$

$$= \sum_{t=1}^{N_r} \mathbb{E}_{y_t|\theta}[\log P(w_{t+1}|y_t)], \tag{11}$$

where the word likelihood at each timestep is obtained by marginalizing out the latent source variable y_t. Unfortunately, direct optimization of Equation 9 is intractable, so we instead learn the objective function through optimizing its variational lower bound given in Equations 10 and 11, obtained from Jensen's inequality.

To estimate the expectation in Equation 11, we use Monte Carlo sampling on the source selector variables $\mathbf{y}$ in the gradient computation. In particular, the Gumbel-Softmax trick is applied to generate discrete samples $\hat{\mathbf{y}}$ from the probability $P(y_t|\mathbf{c}_t^q, \mathbf{c}_t^p, \mathbf{s}_t^r, \mathbf{x}_t^r)$ given by:

$$P(y_t|\cdot) = \text{softmax}(\mathbf{W}^y \cdot [\mathbf{c}_t^q, \mathbf{c}_t^p, \mathbf{s}_t^r, \mathbf{x}_t^r] + \mathbf{b}^y), \tag{12}$$

where $\mathbf{x}_t^r$ is the embedding of the answer word in timestep t, $\mathbf{W}^y$ and $\mathbf{b}^y$ are learnable parameters. The generated samples are fed to $\log P(w_{t+1}|y_t)$ to estimate the expectation.

7 Experiments

We perform quantitative and qualitative analysis of KEAG through experiments. In our experi-

ments, we also study the impact of the integrated knowledge and the ablations of the KEAG model. In addition, we illustrate how natural answers are generated by KEAG with the aid of external knowledge by analyzing a running example.

7.1 Dataset and Evaluation Metrics

Given our objective of generating natural answers by document reading, the MARCO dataset (Nguyen et al., 2016) released by Microsoft is the best fit for benchmarking KEAG and other answer generation methods. We use the latest MARCO V2.1 dataset and focus on the "*Q&A + Natural Language Generation*" task in the evaluation, the goal of which is to provide the best answer available in natural language that could be used by a smart device / digital assistant.

In the MARCO dataset, the questions are user queries issued to the Bing search engine and the contextual passages are from real web documents. The data has been split into a training set (153,725 QA pairs), a dev set (12,467 QA pairs) and a test set (101,092 questions with unpublished answers). Since true answers are not available in the test set, we hold out the dev set for evaluation in our experiments, and test models for each question on its associated passages by concatenating them all together. We tune the hyper-parameters by cross-validation on the training set.

The answers are human-generated and not necessarily sub-spans of the passages, so the official evaluation tool of MARCO uses the metrics BLEU-1 (Papineni et al., 2002) and ROUGE-L (Lin, 2004). We use both metrics for our evaluation to measure the quality of generated answers against the ground truth.

For external knowledge, we use Concept-Net (Speer et al., 2016), one of the most widely used commonsense knowledge bases. Our KEAG is generic and thus can also be applied to other knowledge bases. ConceptNet is a semantic network representing words and phrases as well as the commonsense relationships between them. After filtering out non-English entities and relation types with few facts, we have 2,823,089 fact triples and 32 relation types for the model to consume.

7.2 Implementation Details

In KEAG, we use 300-dimensional pre-trained *Glove* word embeddings (Pennington et al., 2014) for initialization with update during training. The dimension of hidden states is set to 256 for every

Model	Rouge-L	Bleu-1
BiDAF	19.42	13.03
BiDAF+Seq2Seq	34.15	29.68
S-Net	42.71	36.19
S-Net+Seq2Seq	46.83	39.74
QFS	40.58	39.96
VNET	45.93	41.02
gQA	45.75	41.10
KEAG	**51.68**	**45.97**

Table 1: Metrics of KEAG and QA models disregarding knowledge on the MARCO dataset.

LSTM. The fact representation $\mathbf{f}$ has 500 dimensions. The maximum number of related facts N_f is set to be 1000. We use a vocabulary of 50K words (filtered by frequency). Note that the source selector enables KEAG to handle out-of-vocabulary words by generating a word from given text or knowledge.

At both training and test stages, we truncate a passage to 800 words, and limit the length of an answer to 120 words. We train on a single Tesla M40 GPU with the batch size of 16. At test time, answers are generated using beam search with the beam size of 4.

7.3 Model Comparisons

Table 1 compares KEAG with the following state-of-the-art extractive/generative QA models, which do not make use of external knowledge:

1. **BiDAF** (Seo et al., 2017): A multi-stage hierarchical process that represents the context at different levels of granularity, and using the bi-directional attention flow mechanism for answer extraction

2. **BiDAF+Seq2Seq**: A BiDAF model followed by an additional sequence-to-sequence model for answer generation

3. **S-Net** (Tan et al., 2018): An extraction-then-synthesis framework to synthesize answers from extracted evidences

4. **S-Net+Seq2Seq**: An S-Net model followed by an additional sequence-to-sequence model for answer generation

5. **QFS** (Nema et al., 2017): A model that adapts the query-focused summarization model to answer generation

6. **VNET** (Wang et al., 2018): An MRC model that enables answer candidates from different

Model	Rouge-L	Bleu-1
gQA w/ KBLSTM	49.33	42.81
gQA w/ CRWE	49.79	43.35
MHPGM	50.51	44.73
KEAG	**51.68**	**45.97**

Table 2: Metrics of KEAG and knowledge-enriched QA models on the MARCO dataset.

Model	Syntactic	Correct
gQA	3.78	3.54
gQA w/ KBLSTM	3.98	3.62
gQA w/ CRWE	3.91	3.69
MHPGM	4.10	3.81
KEAG	**4.18**	**4.03**

Table 3: Human evaluation of KEAG and state-of-the-art answer generation models. Scores range in $[1, 5]$.

passages to verify each other based on their content representations

7. **gQA** (Mitra, 2017): A generative approach to question answering by incorporating the copying mechanism and the coverage vector

Table 1 shows the comparison of QA models in Rouge-L and Bleu-1. From the table we observe that abstractive QA models (e.g., KEAG) are consistently superior to extractive models (e.g., BiDAF) in answer quality. Therefore, abstractive QA models establish a strong base architecture to be enhanced with external knowledge, which motivates this work. Among the abstractive models, gQA can be viewed as a simplification of KEAG, which generates answer words from passages and the vocabulary without the use of knowledge. In addition, KEAG incorporates a stochastic source selector while gQA does not. The result that KEAG significantly outperforms gQA demonstrates the effectiveness of KEAG's architecture and the benefit of knowledge integration.

Table 2 shows the metrics of KEAG in comparison to those of the following state-of-the-art QA models that are adapted to leveraging knowledge:

1. **gQA w/ KBLSTM** (Yang and Mitchell, 2017): KBLSTM is a neural model that leverages continuous representations of knowledge bases to enhance the learning of recurrent neural networks for machine reading. We plug it into gQA to make use of external knowledge for natural answer generation.

2. **gQA w/ CRWE** (Weissenborn, 2017): CRWE is a reading architecture with dynamic integration of background knowledge based on contextual refinement of word embeddings by leveraging supplementary knowledge. We extend gQA with the refined word embedding for this model.

3. **MHPGM** (Bauer et al., 2018): A multi-hop reasoning QA model which fills in the gaps of inference with commonsense knowledge.

From Table 2, it can be clearly observed that KEAG performs best with the highest Rouge-L and Bleu-1 scores among the knowledge-enriched answer generation models. The major difference between KEAG and the other models is the way of incorporating external knowledge into a model. *gQA w/ KBLSTM* and *gQA w/ CRWE* extend gQA with the module that consumes knowledge, and MHPGM incorporates knowledge with selectively-gated attention while its decoder does not leverage words from knowledge in answer generation. Different from these models, KEAG utilizes two stochastic selectors to determine when to leverage knowledge and which fact to use. It brings additional gains in exploiting external knowledge to generate abstractive answers.

Since neither Rouge-L nor Bleu-1 can measure the quality of generated answers in terms of their correctness and accuracy, we also conduct human evaluation on Amazon Mechanical Turk. The evaluation assesses the answer quality on grammaticality and correctness. We randomly select 100 questions from the dev set, and ask turkers for ratings in a Likert scale ($\in [1, 5]$) on the generated answers.

Table 3 reports the human evaluation scores of KEAG and state-of-the-art answer generation models. The KEAG model surpasses all the others in generating correct answers syntactically and substantively. In terms of syntactic correctness, KEAG and MHPGM both perform well thanks to their architectures of composing answer text and integrating knowledge. On the other hand, KEAG significantly outperforms all compared models in generating substantively correct answers, which demonstrates its power in exploiting external knowledge.

7.4 Ablation Studies

We conduct ablation studies to assess the individual contribution of every component in KEAG. Table 4 reports the performance of the full KEAG

Ablation	Rouge-L	Bleu-1
Full KEAG	**51.68**	**45.97**
✗ supplementary knowledge	49.98	44.59
✗ latent indicators **y**	47.61	42.10
✗ source selector	38.33	36.75

Table 4: Ablation tests of KEAG.

model and its ablations.

We evaluate how much incorporating external knowledge as supplementary information contributes to natural answer generation by removing the supplementary knowledge and the corresponding fact selection module from KEAG's architecture. It can be seen that the knowledge component plays an important role in generating high-quality answers, with a drop to 49.98 on Rouge-L after the supplementary knowledge is removed.

To study the effect of our learning method, we further ablate the latent indicators **y**, which leads to degradation to gQA except that the new model can select answer words from the question source while gQA cannot. Our learning method proves to be effective with a drop of about 5% on Rouge-L and about 6% on Bleu-1 after ablation.

Finally, for ablating the source selector, we have a new model that generates answer words from the vocabulary alone. It results in a significant drop to 38.33 on Rouge-L, confirming its effectiveness in generating natural answers.

7.5 Visualization and Interpretation

The source selector allows us to visualize how every word in an answer is generated from one of the sources of the question, passage, vocabulary and knowledge, which gives us insights about how KEAG works.

Table 5 visualizes a sample QA pair from KEAG and which source every word in the answer is selected from (indicated by the sample value of the source selector variable y_t). As exemplified in the table, the source distribution $P(y_t|\theta)$ varies over decoding timesteps. To answer the question, at each timestep, KEAG first selects a source based on the sample from $P(y_t|\theta)$, followed by generating an answer word from the selected source. It is observed that the in the generated answer the keyword *personality* comes from the knowledge source which relates psychopathy to personality. The answer word *psychopathy* is selected from the question source, which leads to a well-formed answer with a complete sen-

Question	
What's psychopathy?	
Answer with source probabilities	
Question src	Psychopathy is a personality disorder.
Passage src	Psychopathy is a personality disorder.
Vocabulary src	Psychopathy is a personality disorder.
Knowledge src	Psychopathy is a personality disorder.
Answer colored by source	
Psychopathy is a personality disorder.	

Table 5: Visualization of a sample QA pair and the source of individual words in the answer. The **Answer with source probabilities** section displays a heatmap on answer words selected from the question, passage, vocabulary and knowledge, respectively. A slot with a higher source probability is highlighted in darker cyan. The **Answer colored by source** section shows the answer in which every word is colored based on the source it was actually selected from. Words in blue come from the question, red from the passage, green from the vocabulary, and orange from the knowledge. The visualization is best viewed in color.

tence. Another keyword *disorder*, on the other hand, comes from the passage source. This results from reading comprehension of the model on the passage. To generate a final answer in good form, KEAG picks the filler words *is* and *a* as well as the period "." from the vocabulary source. It makes the generated answer semantically correct and comprehensive.

8 Conclusion and Future Work

This paper presents a new neural model KEAG that is designed to bring symbolic knowledge from a knowledge base into abstractive answer generation. This architecture employs the source selector that allows for learning an appropriate tradeoff for blending external knowledge with information from textual context. The related fact extraction and stochastic fact selection modules are introduced to complete an answer with relevant facts.

This work opens up for deeper investigation of answer generation models in a targeted way, allowing us to investigate what knowledge sources are required for different domains. In future work, we will explore even tighter integration of symbolic knowledge and stronger reasoning methods.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.

Yao Fu and Yansong Feng. 2018. Natural answer generation with heterogeneous memory. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 185–195.

Shen Gao, Zhaochun Ren, Yihong Zhao, Dongyan Zhao, Dawei Yin, and Rui Yan. 2019. Product-aware answer generation in e-commerce question-answering. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*.

E.J. Gumbel. 1954. *Statistical theory of extreme values and some practical applications: a series of lectures*. Applied mathematics series. U. S. Govt. Print. Office.

Shizhu He, Cao Liu, Kang Liu, and Jun Zhao. 2017. Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 199–208.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparametrization with gumbel-softmax. In *Proceedings of the International Conference on Learning Representations*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL Workshop: Text Summarization Braches Out 2004*, page 10.

Teng Long, Emmanuel Bengio, Ryan Lowe, Jackie Chi Kit Cheung, and Doina Precup. 2017. World knowledge for reading comprehension: Rare entity prediction with hierarchical lstms using external descriptions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 825–834.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832.

Rajarshee Mitra. 2017. An abstractive approach to question answering. *CoRR*, abs/1711.06238.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*.

Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1063–1072. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS)*.

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*, pages 311–318.

Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. 2016. From freebase to wikidata: The great migration. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1419–1428.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *Proceedings of the International Conference on Learning Representations*.

Robert Speer, Joshua Chin, and Catherine Havasi. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence*.

Chuanqi Tan, Furu Wei, Nan Yang, Bowen Du, Weifeng Lv, and Ming Zhou. 2018. S-net: From answer extraction to answer synthesis for machine reading comprehension. In *AAAI*.

Yizhong Wang, Kai Liu, Jing Liu, Wei He, Yajuan Lyu, Hua Wu, Sujian Li, and Haifeng Wang. 2018. Multi-passage machine reading comprehension with cross-passage answer verification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1918–1927, Melbourne, Australia. Association for Computational Linguistics.

Dirk Weissenborn. 2017. Dynamic integration of background knowledge in neural NLU systems. *CoRR*, abs/1706.02596.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in LSTMs for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1436–1446.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural generative question answering. In *Proceedings of the Workshop on Human-Computer Question Answering*, pages 36–42.

Wanjun Zhong, Duyu Tang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2018. Improving question answering by commonsense-based pre-training. *CoRR*, abs/1809.03568.