

Neural Models for Key Phrase Extraction and Question Generation

Sandeep Subramanian^{♣♠} Tong Wang[♣] Xingdi Yuan[♣] Saizheng Zhang[♠]
Yoshua Bengio^{♠†} Adam Trischler[♣]

♣Microsoft Research, Montréal ♠MILA, Université de Montréal †CIFAR Senior Fellow
sandeep.subramanian.1@umontreal.ca

Abstract

We propose a two-stage neural model to tackle question generation from documents. First, our model estimates the probability that word sequences in a document are ones that a human would pick when selecting candidate answers by training a neural key-phrase extractor on the answers in a question-answering corpus. Predicted key phrases then act as target answers and condition a sequence-to-sequence question-generation model with a copy mechanism. Empirically, our key-phrase extraction model significantly outperforms an entity-tagging baseline and existing rule-based approaches. We further demonstrate that our question generation system formulates fluent, answerable questions from key phrases. This two-stage system could be used to augment or generate reading comprehension datasets, which may be leveraged to improve machine reading systems or in educational settings.

1 Introduction

Question answering and machine comprehension has gained increased interest in the past few years. An important contributing factor is the emergence of several large-scale QA datasets (Rajpurkar et al., 2016; Trischler et al., 2016; Nguyen et al., 2016; Joshi et al., 2017). However, the creation of these datasets is a labour-intensive and expensive process that usually comes at significant financial cost. Meanwhile, given the complexity of the problem space, even the largest QA dataset can still exhibit strong biases in many aspects including question and answer types, domain coverage, linguistic style, etc.

To address this limitation, we propose and evaluate neural models for automatic question-answer pair generation that involves two inter-related components: first, a system to identify candidate answer entities or events (key phrases) within a passage or document (Becker et al., 2012); second, a question generation module to construct questions about a given key phrases. As a financially more efficient and scalable alternative to the human curation of QA datasets, the resulting system can potentially accelerate further progress in the field.

Specifically, We formulate the key phrase extraction component as modeling the probability of potential answers conditioned on a given document, i.e., $P(a|d)$. Inspired by successful work in question answering, we propose a sequence-to-sequence model that generates a set of key-phrase *boundaries*. This model can flexibly select an arbitrary number of key phrases from a document. To teach it to assign high probability to human-selected answers, we train the model on large-scale, crowd-sourced question-answering datasets.

We thus take a purely data-driven approach to understand the priors that humans have when selecting answer candidates, working from the premise that crowdworkers tend to select entities or events that interest them when formulating their own comprehension questions. If this premise is correct, then the growing collection of crowd-sourced question-answering datasets (Rajpurkar et al., 2016; Trischler et al., 2016) can be harnessed to learn models for key phrases of interest to human readers.

Given a set of extracted key phrases, we then approach the question generation component by modeling the conditional probability of a question given a document-answer pair, i.e., $P(q|a, d)$. To this end, we use a sequence-to-sequence model with attention (Bahdanau et al., 2014) and

the pointer-softmax mechanism (Gulcehre et al., 2016). This component is also trained to maximize the likelihood of questions estimated on a QA dataset. When training this component, the model sees the ground truth answers from the dataset.

Empirically, our proposed model for key phrase extraction outperforms two baseline systems by a significant margin. We support these quantitative findings with qualitative examples of generated question-answer pairs given documents.

2 Related Work

2.1 Key Phrase Extraction

An important aspect of question generation is identifying which elements of a given document are important or interesting to inquire about. Existing studies formulate key-phrase extraction in two steps. In the first, lexical features (e.g., part-of-speech tags) are used to extract a key-phrase candidate list exhibiting certain types (Liu et al., 2011; Wang et al., 2016; Le et al., 2016; Yang et al., 2017). In the second, ranking models are often used to select a phrase from among the candidates. Medelyan et al. (2009); Lopez and Romary (2010) used bagged decision trees, while Lopez and Romary (2010) used a Multi-Layer Perceptron (MLP) and Support Vector Machine to perform binary classification on the candidates. Mihalcea and Tarau (2004); Wan and Xiao (2008); Le et al. (2016) scored key phrases using *PageRank*. Heilman and Smith (2010b) asked crowdworkers to rate the acceptability of computer-generated *natural language* questions as quiz questions, and Becker et al. (2012) solicited quality ratings of text chunks as potential gaps for *Cloze-style* questions.

These studies are closely related to our proposed work by the common goal of modeling the distribution of key phrases given a document. The major difference is that previous studies begin with a prescribed list of candidates, which might significantly bias the distribution estimate. In contrast, we adopt a dataset that was originally designed for question answering, where crowdworkers presumably tend to pick entities or events that interest them most. We postulate that the resulting distribution, learned directly from data, is more likely to reflect the true relevance of potential answer phrases.

Recently, Meng et al. (2017) proposed a generative model for key phrase prediction with an

encoder-decoder framework that is able both to generate words from a vocabulary and point to words from the document. Their model achieved state-of-the-art results on multiple keyword-extraction datasets. This model shares certain similarities with our key phrase extractor, i.e., using a single neural model to learn the probabilities that words are key phrases. Since their focus was on a hybrid abstractive-extractive task in contrast to the purely extractive task in this work, a direct comparison between works is difficult.

Yang et al. (2017) used rule-based methods to extract potential answers from unlabeled text, and then generated questions given documents and extracted answers using a pre-trained question generation model. The model-generated questions were then combined with human-generated questions for training question answering models. Experiments showed that question answering models can benefit from the augmented data provided by their approach.

2.2 Question Generation

Automatic question generation systems are often used to alleviate (or eliminate) the burden of human generation of questions to assess reading comprehension (Mitkov and Ha, 2003; Kunichika et al., 2004). Various NLP techniques have been adopted in these systems to improve generation quality, including parsing (Heilman and Smith, 2010a; Mitkov and Ha, 2003), semantic role labeling (Lindberg et al., 2013), and the use of lexicographic resources like *WordNet* (Miller, 1995; Mitkov and Ha, 2003). However, the majority of the proposed methods resort to simple, rule-based techniques such as template-based slot filling (Lindberg et al., 2013; Chali and Golestanirad, 2016; Labutov et al., 2015) or syntactic transformation heuristics (Agarwal and Mannem, 2011; Ali et al., 2010) (e.g., subject-auxiliary inversion, (Heilman and Smith, 2010a)). These techniques generally do not capture the diversity of human generated questions.

To address this limitation, end-to-end-trainable neural models have recently been proposed for question generation in both vision (Mostafazadeh et al., 2016) and language. For the latter, Du et al. (2017) used a sequence-to-sequence model with an attention mechanism derived from the encoder states. Yuan et al. (2017) proposed a similar architecture but further improved model per-

formance with policy gradient techniques. Wang et al. (2017) proposed a generative model that learns jointly to generate questions or answers from documents.

3 Model Description

3.1 Notations

Several components introduced in the following sections share the same model architecture for encoding text sequences. The common notations are explained in this section.

Unless otherwise specified, w refers to word tokens, e to word embeddings and h to the annotation vectors (also commonly referred to as hidden states) produced by an RNN. Superscripts specify the source of a word, e.g., d for documents, p for key phrases, a for (gold) answers, and q for questions. Subscripts index the position inside a sequence. For example, e_i^d is the embedding vector for the i -th token in the document.

A sequence of words are often encoded into *annotation vectors* (denoted h) by applying a bidirectional LSTM encoder to the corresponding sequence of word embeddings. For example, $h_j^q = \text{LSTM}(e_j^q, h_{j-1}^q)$ is the annotation vector for the j -th word in a question.

3.2 Key Phrase Extraction

In this section, we describe a simple baseline as well as two proposed neural models for extracting key phrases (answers) from documents.

3.2.1 Entity Tagging Baseline

As our first baseline, we use *spaCy*¹ to predict all entities in a document as relevant key phrases (call this model *ENT*). This is motivated by the fact that entities constitute the largest proportion (over 50%) of answers in the SQuAD dataset (Rajpurkar et al., 2016). Entities includes dates (*September 1967*), numeric entities (*3*, *five*), people (*William Smith*), locations (*the British Isles*), and other named concepts (*Buddhism*).

3.2.2 Neural Entity Selection

The baseline model above naïvely selects all entities as candidate answers. One pitfall is that it exhibits high recall at the expense of precision (Table 1), since not all entities lead to interesting questions. We first attempt to address this with a neural entity selection model (NES) that selects a

subset of entities from a list of candidates provided by our ENT baseline. Our neural model takes as input a document (i.e., a sequence of words), $D = (w_1^d, \dots, w_{n_d}^d)$, and a list of n_e entities as a sequence of $(start, end)$ locations within the document, $E = ((e_1^{start}, e_1^{end}), \dots, (e_{n_e}^{start}, e_{n_e}^{end}))$. The model is then trained on the binary classification task of predicting whether an entity overlaps with any of the human-provided answers.

Specifically, we maximize $\sum_i^{n_e} \log(P(e_i|D))$. We parameterize $P(e_i|D)$ using a three-layer multilayer perceptron (MLP) that takes as input the concatenation of three vectors $\langle h_{n_d}^d; h_{avg}^d; h_{e_i} \rangle$. Here, h_{avg}^d and $h_{n_d}^d$ are the average and the final state of the document annotation vectors, respectively, and h_{e_i} is the average of the annotation vectors corresponding to the i -th entity (i.e., $h_{e_i^{start}}^d, \dots, h_{e_i^{end}}^d$).

During inference, we select the top k entities with highest likelihood under our model. We use $k = 6$ in our experiments as determined by hyperparameter search.

3.2.3 Pointer Networks

While a significant fraction of answers in QA datasets like SQuAD are entities, entities alone may be insufficient for detecting different aspects of a document. Many documents are entity-less, and entity taggers may fail to recognize some entities. To this end, we build a neural model that is trained from scratch to extract all human-selected answer phrases in a particular document. We parameterize this model as a pointer network (Vinyals et al., 2015) trained to point sequentially to the start and end *locations* of all labeled answers in a document. An autoregressive decoder LSTM augmented with an attention mechanism is then trained to point (attend) to all of the start and end locations of answers *from left to right*, conditioned on the annotation vectors (extracted in the same fashion as in the NES model), via an attention mechanism. We add a special termination token to the document and train the decoder to attend to it once it has generated all key phrases. This enables the model to extract variable numbers of key phrases depending on the input document. This is in contrast to the work of Meng et al. (2017), where a fixed number of key phrases is generated per document.

A pointer network is an extension of sequence-to-sequence models (Sutskever et al., 2014),

¹<https://spacy.io/docs/usage/entity-recognition>

where the target sequence consists of positions in the source sequence. An autoregressive decoder RNN is trained to attend to these positions in the input conditioned on an encoding of the input produced by an encoder RNN. We denote the decoder’s annotation vectors as $(\mathbf{h}_1^p, \mathbf{h}_2^p, \dots, \mathbf{h}_{2n_a-1}^p, \mathbf{h}_{2n_a}^p)$, where n_a is the number of answer key phrases, \mathbf{h}_1^p and \mathbf{h}_2^p correspond to the start and end annotation vectors for the first answer key phrase, and so on. We parameterize $P(w_i^d = \text{start} | \mathbf{h}_1^p \dots \mathbf{h}_j^p, \mathbf{h}^d)$ and $P(w_i^d = \text{end} | \mathbf{h}_1^p \dots \mathbf{h}_j^p, \mathbf{h}^d)$ using the general attention mechanism (Luong et al., 2015) between the decoder and encoder annotation vectors,

$$P(w_i^d | \mathbf{h}_1^p \dots \mathbf{h}_j^p, \mathbf{h}^d) = \text{softmax}(W_1 \mathbf{h}_j^p \cdot \mathbf{h}^d),$$

where W_1 is a learned parameter matrix. The inputs at each step of the decoder are words from the document that correspond to the start and end locations pointed to by the decoder.

During inference, we employ a decoding strategy that greedily picks the best location from the softmax vector at every step, then post process results to remove duplicate key phrases. Since the output sequence is relatively short, we observed similar performances when using greedy decoding and beam search.

We also experimented with a BIO tagging model using an LSTM-CRF (Lample et al., 2016) but were unable to make the model predict anything except “O” for every token.

3.3 Question Generation

The question generation model adopts a sequence-to-sequence framework (Sutskever et al., 2014) with an attention mechanism (Bahdanau et al., 2014) and a pointer-softmax decoder (Gulcehre et al., 2016). We make use of the pointer-softmax mechanism since it lets us take advantage of the inherent nature of RC datasets re-using words in the document when framing questions. Our setup for this module is identical to (Yuan et al., 2017). It takes a document $w_{1:n_d}^d$ and an answer $w_{1:n_a}^a$ as input, and outputs a question $\hat{w}_{1:n_q}^q$.

An input word $w_i^{\{d,a\}}$ is represented by concatenating its word embedding e_i with character-level embedding e_{ch_i} . Each character in the alphabet receives an embedding vector, and e_{ch_i} is the final state of a bi-LSTM running over the embedding vectors corresponding to the character sequence of the word.

To leverage the extractive nature of answers in SQuAD, we encode an answer using the document annotation vectors at the answer-word positions. Specifically, if an answer phrase $w_{1:n}^a$ occupies the document span $w_{a_1:a_n}^d$, we first encode the corresponding document annotation vectors with a *condition aggregation* BiLSTM into $\mathbf{h}'_{1:n}$. The concatenation of the final state \mathbf{h}'_n with the answer annotation vector \mathbf{h}_n^a as the answer representation.

The RNN decoder employs a pointer-softmax module (Gulcehre et al., 2016). At each step of the generation process, the decoder decides adaptively whether to (a) generate from the decoder vocabulary or (b) point to a word in the source sequence (the document) and copy over. The pointer-softmax decoder thus has two components — a pointer attention mechanism and a generative decoder.

The subsequent mathematical notation deviates from the previous notation slightly, we use (t) as the superscript. In the **pointing decoder**, recurrence is implemented with two cascading LSTM cells c_1 and c_2 :

$$\mathbf{s}_1^{(t)} = c_1(\mathbf{y}^{(t-1)}, \mathbf{s}_2^{(t-1)}) \quad (1)$$

$$\mathbf{s}_2^{(t)} = c_2(\mathbf{v}^{(t)}, \mathbf{s}_1^{(t)}), \quad (2)$$

where $\mathbf{s}_1^{(t)}$ and $\mathbf{s}_2^{(t)}$ are the recurrent states, $\mathbf{y}^{(t-1)}$ is the embedding of decoder output from the previous time step, and $\mathbf{v}^{(t)}$ is the context vector, which is the sum of the document annotations \mathbf{h}_i^d weighted by the document attention $\alpha_i^{(t)}$ (Equation (3)):

$$\mathbf{v}^{(t)} = \sum_{i=1}^n \alpha_i^{(t)} \mathbf{h}_i^d.$$

At each time step t , the pointing decoder computes a distribution $\alpha^{(t)}$ over the document word positions (i.e., a document attention, Bahdanau et al. 2014). Each element is defined as:

$$\alpha_i^{(t)} = f(\mathbf{h}_i^d, \mathbf{h}^a, \mathbf{s}_1^{(t-1)}), \quad (3)$$

where f is a two-layer MLP with *tanh* and *softmax* activation, respectively.

The **generative decoder**, on the other hand, defines a distribution over a prescribed decoder vocabulary with a two-layer MLP g :

$$\mathbf{o}^{(t)} = g(\mathbf{y}^{(t-1)}, \mathbf{s}_2^{(t)}, \mathbf{v}^{(t)}, \mathbf{h}^a).$$

Table 1: Model evaluation on key phrase extraction

| Models | Validation | | | Test | | |
|---------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | $F1_{MS}$ | Prec. | Rec. | $F1_{MS}$ | Prec. | Rec. |
| SQuAD | | | | | | |
| H&S | - | - | - | 0.292 | 0.252 | 0.403 |
| ENT | 0.308 | 0.249 | 0.523 | 0.347 | 0.295 | 0.547 |
| NES | 0.334 | 0.335 | 0.354 | 0.362 | 0.375 | 0.380 |
| PtrNet | 0.352 | 0.387 | 0.337 | 0.404 | 0.448 | 0.387 |
| NewsQA | | | | | | |
| ENT | 0.187 | 0.127 | 0.491 | 0.183 | 0.125 | 0.479 |
| PtrNet | 0.452 | 0.480 | 0.444 | 0.435 | 0.467 | 0.427 |

Pointer-softmax is implemented by interpolating the generative and the pointing distributions:

$$P(\hat{w}_t) \sim s^{(t)}\alpha^{(t)} + (1 - s^{(t)})\mathbf{o}^{(t)},$$

where $s^{(t)}$ is a switch scalar computed at each time step by a three-layer MLP h :

$$s^{(t)} = h(s_2^{(t)}, \mathbf{v}^{(t)}, \alpha^{(t)}, \mathbf{o}^{(t)}).$$

The first two layers of h use *tanh* activation with highway connections, and the final layer uses *sigmoid* activation.²

4 Experiments and Results

4.1 Datasets

We conduct our experiments on the SQuAD (Rajpurkar et al., 2016) and NewsQA (Trischler et al., 2016) datasets. These are machine comprehension corpora consisting of over 100k crowd-sourced question-answer pairs. SQuAD contains 536 paragraphs from Wikipedia while NewsQA was created on 12,744 news articles. Simple preprocessing is performed, including lower-casing and word tokenization using *NLTK*. Since the test split of SQuAD is hidden from the public, we use 5,158 question-answer pairs (self-contained in 23 Wikipedia articles) from the training set for development, and use the official development data to report test results.

4.2 Implementation Details

We train all models by stochastic gradient descent, with a minibatch size of 32, using the ADAM optimizer.

²We also attach the entropy of the softmax distributions to the input of the final layer, postulating that this guides the switching mechanism by indicating the confidence of pointing vs generating. We observed an improvement in question quality with this modification.

4.2.1 Key Phrase Extraction

Key phrase extraction models use pretrained, 300-dimensional word embeddings generated using a word2vec extension (Ling et al., 2015) and the *English Gigaword 5* corpus. We used bidirectional LSTMs of 256 dimensions (128 forward and backward) to encode the document and an LSTM of 256 dimensions as our decoder in the pointer network. A dropout rate of 0.5 was used at the output of every layer in the network.

4.2.2 Question Generation

The question decoder uses a vocabulary of the 2000 most frequent words in the training data (questions only). This limited vocabulary is possible because the question generator may copy over out-of-vocabulary words from the document with its Pointer-Softmax mechanism. The decoder embedding matrix is initialized with 300-dimensional *GloVe* vectors (Pennington et al., 2014), and dimensionality of the character representations is 32. The number of hidden units is 384 for both the encoder and decoder RNN cells. Dropout is applied at a rate of 0.3 to all embedding layers as well as between the hidden states in the encoder/decoder RNNs across time steps.

4.3 Quantitative Evaluation of Key Phrase Extraction

Since each key phrase is itself a multi-word unit, we believe that a naïve, word-level *F1* that considers an entire key phrase as a single unit is not well suited to this evaluation. We therefore propose an extension of the SQuAD *F1* metric (for a single answer span) to multiple spans within a document, which we call the *multi-span F1 score*.

This metric is calculated as follows. Given the predicted phrase \hat{e}_i and a gold phrase e_j , we first construct a pairwise, token-level *F1* score matrix of elements $f_{i,j}$ between the two phrases \hat{e}_i and e_j . Max-pooling along the gold-label axis essentially assesses the precision of each prediction, with partial matches accounted for by the pairwise *F1* (identical to evaluation of a single answer in SQuAD) in the cells: $p_i = \max_j(f_{i,j})$. Analogously, the recall for label e_j can be computed by max-pooling along the prediction axis: $r_j = \max_i(f_{i,j})$. We define the *multi-span F1 score* using the mean precision $\bar{p} = \text{avg}(p_i)$ and recall $\bar{r} = \text{avg}(r_j)$:

$$F1_{MS} = \frac{2\bar{p} \cdot \bar{r}}{(\bar{p} + \bar{r})}.$$

Table 2: Qualitative examples of detected key phrases and generated questions.

| | | |
|------|---|--|
| Doc. | <i>inflammation is one of the first responses of the immune system to infection . the symptoms of inflammation are redness , swelling , heat , and pain , which are caused by increased blood flow into tissue . inflammation is produced by eicosanoids and cytokines , which are released by injured or infected cells . eicosanoids include prostaglandins that produce fever and the dilation of blood vessels associated with inflammation , and leukotrienes that attract certain white blood cells (leukocytes)</i> ... | |
| Q-A | <i>by eicosanoids and cytokines — who is inflammation produced by ?</i> | <i>of the first responses of the immune system to infection — what is inflammation one of ?</i> |
| Q-A | <i>leukotrienes — what can attract certain white blood cells ?</i> | <i>eicosanoids and cytokines — what are bacteria produced by ?</i> |
| Q-A | <i>inflammation — what is one of the first responses the immune system has to infection ?</i> | <i>eicosanoids and cytokines — what compounds are released by injured or infected cells , triggering inflammation ?</i> |
| Doc. | <i>research shows that student motivation and attitudes towards school are closely linked to student-teacher relationships . enthusiastic teachers are particularly good at creating beneficial relations with their students . their ability to create effective learning environments that foster student achievement depends on the kind of relationship they build with their students . useful teacher-to-student interactions are crucial in linking academic success with personal achievement . here , personal success is a student 's internal goal of improving himself , whereas academic success includes the goals he receives from his superior . a teacher must guide his student in aligning his personal goals with his academic goals . students who receive this positive influence show stronger self-confidenche and greater personal and academic success than those without these teacher interactions .</i> | |
| Q-A | <i>research — what shows that student motivation and attitudes towards school are closely linked to student-teacher relationships ?</i> | <i>useful teacher-to-student interactions — what are crucial in linking academic success with personal achievement ?</i> |
| Q-A | <i>to student-teacher relationships — what does research show that student motivation and attitudes towards school are closely linked to ?</i> | <i>that student motivation and attitudes towards school are closely linked to student-teacher relationships — what does research show to ?</i> |
| Q-A | <i>student-teacher relationships — what are the student motivation and attitudes towards school closely linked to ?</i> | <i>enthusiastic teachers — who are particularly good at creating beneficial relations with their students ?</i> |
| Q-A | <i>teacher-to-student interactions — what is crucial in linking academic success with personal achievement ?</i> | <i>a teacher — who must guide his student in aligning his personal goals ?</i> |
| Q-A | <i>student-teacher relationships — what is student motivation about school linked to ?</i> | <i>beneficial — what type of relationships do enthusiastic teachers cause ?</i> |
| Q-A | <i>aligning his personal goals with his academic goals . — what should a teacher guide a student in ?</i> | <i>student motivation and attitudes towards school — what is strongly linked to good student-teacher relationships ?</i> |
| Doc. | <i>the yuan dynasty was the first time that non-native chinese people ruled all of china . in the historiography of mongolia , it is generally considered to be the continuation of the mongol empire . mongols are widely known to worship the eternal heaven</i> ... | |
| Q-A | <i>the first time — what was the yuan dynasty that non-native chinese people ruled all of china ?</i> | <i>the yuan dynasty — what was the first time that non-native chinese people ruled all of china ?</i> |
| Q-A | <i>the mongol empire — the yuan dynasty is considered to be the continuation of what ?</i> | <i>worship the eternal heaven — what are mongols widely known to do in historiography of mongolia ?</i> |
| Q-A | <i>non-native chinese people — the yuan was the first time all of china was ruled by whom ?</i> | <i>the eternal heaven — what did mongols worship ?</i> |
| Doc. | <i>on july 31 , 1995 , the walt disney company announced an agreement to merge with capital cities/abc for \$ 19 billion in 1998 , abc premiered the aaron sorkin-created sitcom sports night , centering on the travails of the staff of a sportscenter-style sports news program ; despite earning critical praise and multiple emmy awards , the series was cancelled in 2000 after two seasons .</i> | |
| Q-A | <i>an agreement to merge with capital cities/abc for \$19 billion — what did the walt disney company announce on july 31 , 1995 ?</i> | <i>the walt disney company — what announced an agreement to merge with capital cities/abc for \$19 billion on july 31 , 1995 ?</i> |
| Q-A | <i>2000 — in what year was the aaron sorkin-created sitcom sports night cancelled ?</i> | <i>walt disney company — who announced an agreement to merge with capital cities/abc for \$ 19 billion ?</i> |
| Q-A | <i>july 31 , 1995 — when was the disney and abc merger first announced ?</i> | <i>sports night — what aaron sorkin created show did abc debut in 1998 ?</i> |

Note that existing evaluations (e.g., that of Meng et al. (2017)) can be seen as the above computation performed on the matrix of exact match scores between predicted and gold key phrases. By using token-level F1 scores between phrase pairs, we allow fuzzy matches.

Our evaluation of key phrase extraction systems by this metric is presented in Table 1. We compare answer phrases extracted by the method of Heilman and Smith (2010a) (henceforth referred to as H&S),³ our baseline entity tagger, the neural

³<http://www.cs.cmu.edu/ark/mheilman/questions/>

entity selection module, and the pointer network. As expected, the entity tagging baseline achieves the best recall, likely by over-generating candidate answers. The NES model, on the other hand, exhibits much better precision and consequently outperforms the entity tagging baseline significantly in *F1*. This trend persists when comparing the NES model and the pointer network. The H&S model exhibits high recall but lacks precision, similar to the baseline entity tagger. This is not surprising since that model is not trained on SQuAD’s answer-phrase distribution.

4.4 Qualitative Evaluation of Key Phrase Extraction

Qualitatively, we observe that the entity-based models have a strong bias for numeric types, which often fail to capture interesting information in a document. We also notice that entity-based systems tend to select the central topical entity as answer, which does not match the distribution of answers typically selected by humans. For example, given a Wikipedia article on Kenya stating that *agriculture is the second largest contributor to kenya’s gross domestic product (gdp)*, entity-based systems propose *kenya* as an answer phrase. This leads to the (low-quality) question *what country is nigeria’s second largest contributor to?*⁴ Given the same document, the pointer model picked *agriculture* as the answer and asked *what is the second largest contributor to kenya’s gross domestic product?*

4.5 Quantitative Evaluation of QA pairs

We can quantitatively evaluate our question generation module by conditioning it on gold answers from the SQuAD development set. We can then use standard automatic evaluation metrics for generative models of text such as BLEU. Our question generation model evaluated in such a manner yields 10.4 BLEU₄.

However, there can exist a many possible ways to formulate a question given the same answer. BLEU thus becomes a less desirable metric by penalizing any generation that does not closely match (lexically) the reference question. To address this issue, we propose to evaluate a generated question by employing a pre-trained QA model. Specifically, suppose question \hat{q} is generated from

⁴Since the answer word *kenya* cannot appear in the generated question, the decoder produced a similar word *nigeria* instead.

document d and answer a , and the pre-trained QA model outputs answer \hat{a} given the input d and \hat{q} . If the QA model is assumed to be able to answer the gold question q with the gold answer a , then the F1 score between a and \hat{a} may serve as a proxy to the *semantic* equivalence between q and \hat{q} — regardless of the amount of word/n-gram overlap between q and \hat{q} .

Quantitatively, a match-LSTM model (Wang and Jiang, 2016) pre-trained on gold squad question/answer pairs achieves an F1 score of 72.4% on our generated questions in comparison to 73.8% on the SQuAD dev set.

In addition to the automatic evaluation metrics, we also undertook a human evaluation of generated questions and answers.

4.6 Qualitative Evaluation of QA pairs

We present several answer-extraction and question-generation examples in Table 2. Each example contains a document and three corresponding QA pairs, generated respectively by H&S, by our two-stage framework, and by the original SQuAD crowdworkers.

We now discuss the relative qualities of QA pairs from each synthetic method.

H&S Key phrases selected by the H&S model are structurally distinct from the PtrNet and human-generated answers. For example, they may start with prepositions, such as *of*, *by*, and *to*, or consist of very long phrases like *that student motivation and attitudes towards school are closely linked to student-teacher relationships*. As seen in Figure 1, these key phrases may also contain vague phrases such as “this theory”, “some studies”, “a person”, etc., which renders them less natural for question generation. The H&S question generator appears to produce a few ungrammatical sentences, e.g., *the first time – what was the yuan dynasty that non-native chinese people ruled all of china?*

Our system Since our key phrase extractor was trained on SQuAD, the selected key phrases more closely resemble gold SQuAD answers. However, sometimes the generated questions do not target the extracted answers, eg, *eicosanoids and cytokines — what are bacteria produced by?* (first document in Table 2). Interestingly, our model is sometimes able to resolve coreferent entities. For instance, to generate *the mongol empire — the*

| | |
|------|---|
| Doc. | <i>following the utilitarian principle of seeking the greatest good for the greatest number –UNK– economic inequality is problematic . a house that provides less utility to a millionaire as a summer home than it would to a homeless family of five , is an example of reduced “ distributive efficiency ” within society , that decreases marginal utility of wealth and thus the sum total of personal utility . an additional dollar spent by a poor person will go to things providing a great deal of utility to that person , such as basic necessities like food , water , and healthcare ; while , an additional dollar spent by a much richer person will very likely go to luxury items providing relatively less utility to that person . thus , the marginal utility of wealth per person (“ the additional dollar ”) decreases as a person becomes richer . from this standpoint , for any given amount of wealth in society , a society with more equality will have higher aggregate utility . some studies have found evidence for this theory , noting that in societies where inequality is lower , population-wide satisfaction and happiness tend to be higher .</i> |
| Doc. | <i>oxygen gas can also be produced through electrolysis of water into molecular oxygen and hydrogen . dc electricity must be used : if ac is used , the gases in each limb consist of hydrogen and oxygen in the explosive ratio 2:1 . contrary to popular belief , the 2:1 ratio observed in the dc electrolysis of acidified water does not prove that the empirical formula of water is h2o unless certain assumptions are made about the molecular formulae of hydrogen and oxygen themselves . a similar method is the electrocatalytic o2 evolution from oxides and oxoacids . chemical catalysts can be used as well , such as in chemical oxygen generators or oxygen candles that are used as part of the life-support equipment on submarines , and are still part of standard equipment on commercial airliners in case of depressurization emergencies . another air separation technology involves forcing air to dissolve through ceramic membranes based on zirconium dioxide by either high pressure or an electric current , to produce nearly pure o2 gas .</i> |
| Doc. | <i>since forces are perceived as pushes or pulls , this can provide an intuitive understanding for describing forces . as with other physical concepts (e.g. temperature) , the intuitive understanding of forces is quantified using precise operational definitions that are consistent with direct observations and compared to a standard measurement scale . through experimentation , it is determined that laboratory measurements of forces are fully consistent with the conceptual definition of force offered by newtonian mechanics .</i> |
| Doc. | <i>on april 20 , kennedy sent a memo to vice president lyndon b. johnson , asking johnson to look into the status of america ’s space program , and into programs that could offer nasa the opportunity to catch up . johnson responded approximately one week later , concluding that “ we are neither making maximum effort nor achieving results necessary if this country is to reach a position of leadership . ” his memo concluded that a manned moon landing was far enough in the future that it was likely the united states would achieve it first .</i> |

Figure 1: A comparison of key phrase extraction methods. Red phrases are extracted by the pointer network, violet by H&S, green by the baseline, brown correspond to squad gold answers and cyan indicates an overlap between the pointer model and squad gold questions. The last paragraph is an exception where **lyndon b. johnson** and **april 20** are extracted by H&S as well as the baseline model.

yuan dynasty is considered to be the continuation of what ? the model must resolve the pronoun it to yuan dynasty in it is generally considered to be the continuation of the mongol empire (third document in Table 2).

4.7 Human Evaluation Studies

We carried out human evaluations on the question generation module in isolation as well as in conjunction with the key phrase extraction module.

Evaluating the ability of the Question Generation Module to transfer to new settings We asked crowdworkers part of an internal evaluation system to evaluate two different aspects of questions generated by our module - fluency and correctness. Our system was provided Internet articles and candidate answers selected from an internal search engine thereby evaluating the model’s ability to generalize from simple RC datasets to the real world. For fluency evaluations, they were asked whether the generated questions sounded natural (ignoring semantics) with scores of 0/1/2 corresponding to “No”, “Somewhat” and “Yes”. 17.5% were labeled 0, 22.7% were labeled 1 and 59.8% were labeled 2. For correctness evaluations, annotators were asked if the given answer was the correct answer for the given question. 64.4% of questions were labeled incorrect, leav-

ing 35.6% labeled as correct. This particular evaluation differs slightly from others with regard to the module used (it was trained a combination of SQuAD + NewsQA + TriviaQA (Joshi et al., 2017)). Also the documents and answers used provided via an internal tool. 1,302 annotations were collected.

Comparison to human generated questions

We present annotators with documents from SQuAD’s official development set and two sets of question-answer pairs, one from our model (machine generated) and the other from SQuAD (human generated). Annotators are then asked to identify which question-answer pair is machine generated. The order in which the pairs appear is randomized across examples. Annotators are free to use any criterion to make a distinction, such as poor grammar, the answer phrase not correctly answering the generated question, unnatural answer phrases, etc.

We presented 14 annotators with a total of 740 documents, each with 2 corresponding QA pairs. Annotators identified the machine generated pairs 77.8% of the time with a standard deviation of 8.34%.

Implicit comparison to H&S To compare our system to existing methods (H&S), we orchestrate

an implicit comparison grounded in human generated QA pairs from SQuAD. We present human annotators with a document and two QA pairs – one that comes from the true development set and the other from either our system or H&S, at random. Annotators are not told that there are two different models generating QA pairs. As above, annotators are asked to identify which QA pair is human generated and which is synthetic.

We presented a single annotator with 100 documents, each with two QA pairs. For 45 documents, the synthetic QA pair came from our model; for the remaining 55, the synthetic pair was from H&S. The annotator distinguished correctly between our system’s output and the human-generated pair in 30 cases (66.7%), and did so in 45 cases (81.8%) for H&S. This experiment suggests that our system’s generated QA pairs are less distinguishable from human QA pairs.

Comparison to H&S In a more direct evaluation, we present annotators with documents from the SQuAD development set along with one QA pair generated by the H&S model and one generated by ours. We then ask annotators which QA pair they prefer.

We presented the same single annotator with 200 such examples. In 107 cases (53.5%), the annotator preferred the pair generated by our model. This suggests that, without human generated QA pairs for comparison, the annotator considers the two models’ outputs to be roughly equal in quality.

5 Conclusion

We propose a two-stage framework to tackle the problem of question generation from documents. First, we use a question answering corpus to train a neural model to estimate the distribution of key phrases that are likely to be picked by humans to ask questions about. We present two neural models, one that ranks entities proposed by an entity tagging system, and another that points to keyphrase start and end boundaries with a pointer network. When compared to an entity tagging baseline, the proposed models exhibit significantly better results.

We adopt a sequence-to-sequence model to generate questions conditioned on the key phrases selected in the framework’s first stage. Our question generator is inspired by an attention-based translation model, and uses the pointer-softmax mech-

anism to dynamically switch between copying a word from the document and generating a word from a vocabulary. Qualitative examples show that the generated questions exhibit both syntactic fluency and semantic relevance to the conditioning documents and answers, and appear useful for assessing reading comprehension in educational settings. In future work we will investigate fine-tuning the two-stage framework end to end. Another interesting direction is to explore abstractive key-phrase extraction.

6 Acknowledgements

The authors would like to thank the reviewers for their valuable feedback.

References

- Manish Agarwal and Prashanth Mannem. 2011. Automatic gap-fill question generation from text books. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 56–64. Association for Computational Linguistics.
- Husam Ali, Yllias Chali, and Sadid A Hasan. 2010. Automation of question generation from sentences. In *Proceedings of QG2010: The Third Workshop on Question Generation*, pages 58–67.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. [Neural machine translation by jointly learning to align and translate](#). *CoRR*, abs/1409.0473.
- Lee Becker, Sumit Basu, and Lucy Vanderwende. 2012. Mind the gap: learning to choose gaps for question generation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 742–751. Association for Computational Linguistics.
- Yllias Chali and Sina Golestanirad. 2016. Ranking automatically generated questions using common human queries. In *The 9th International Natural Language Generation conference*, page 217.
- Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*.
- Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. *arXiv preprint arXiv:1603.08148*.
- Michael Heilman and Noah A Smith. 2010a. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010*

- Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617. Association for Computational Linguistics.
- Michael Heilman and Noah A Smith. 2010b. Rating computer-generated questions with mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 35–40. Association for Computational Linguistics.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.
- Hidenobu Kunichika, Tomoki Katayama, Tsukasa Hishida, and Akira Takeuchi. 2004. Automated question generation methods for intelligent english learning systems and its evaluation. In *Proc. of ICCE*.
- Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *ACL (1)*, pages 889–898.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.
- Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu. 2016. Unsupervised keyphrase extraction: Introducing new kinds of words to keyphrases. *29th Australasian Joint Conference, Hobart, TAS, Australia, December 5-8, 2016*.
- David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. *ENLG 2013*, page 105.
- Wang Ling, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *HLT-NAACL*, pages 1299–1304.
- Zhiyuan Liu, Xinxiong Chen, Yabin Zheng, and Maosong Sun. 2011. Automatic keyphrase extraction by bridging vocabulary gap. *the Fifteenth Conference on Computational Natural Language Learning*.
- Patrice Lopez and Laurent Romary. 2010. Humb: Automatic key term extraction from scientific articles in grobidp. *the 5th International Workshop on Semantic Evaluation*.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Olena Medelyan, Eibe Frank, and Ian H Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. *Empirical Methods in Natural Language*.
- Rui Meng, Sanqiang Zhao, shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *the 55th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. *Empirical Methods in Natural Language*.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2*, pages 17–22. Association for Computational Linguistics.
- Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating natural questions about an image. *arXiv preprint arXiv:1603.06059*.
- 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. *arXiv preprint arXiv:1611.09268*.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. [Glove: Global vectors for word representation](#). In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.
- Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordani, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *2nd Workshop on Representation Learning for NLP*.
- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.
- Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. *AAAI*.

- Minmei Wang, Bo Zhao, and Yihua Huang. 2016. Ptr: Phrase-based topical ranking for automatic keyphrase extraction in scientific publications. *23rd International Conference, ICONIP 2016*.
- Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.
- Tong Wang, Xingdi Yuan, and Adam Trischler. 2017. A joint model for question answering and question generation. *1st Workshop on Learning to Generate Natural Language*.
- Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William W. Cohen. 2017. Semi-supervised qa with generative domain-adaptive nets. In *the 55th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordani, Philip Bachman, Sandeep Subramanian, Saizheng Zhang, and Adam Trischler. 2017. Machine comprehension by text-to-text neural question generation. *2nd Workshop on Representation Learning for NLP*.