Generating Question-Answer Hierarchies

Kalpesh Krishna & Mohit Iyyer
College of Information and Computer Sciences
University of Massachusetts Amherst
{kalpesh,miyyer}@cs.umass.edu

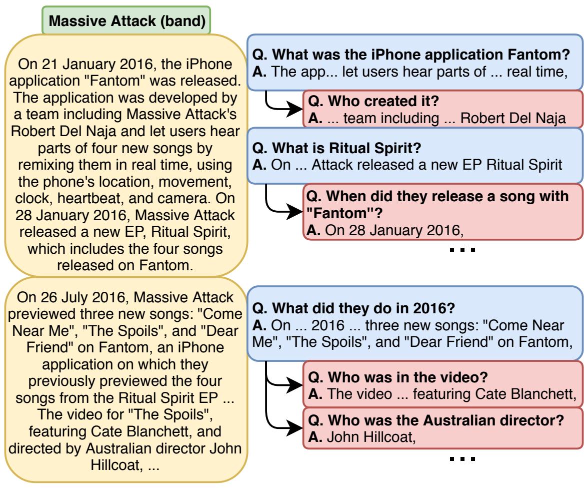
http://squash.cs.umass.edu/

Abstract

The process of knowledge acquisition can be viewed as a question-answer game between a student and a teacher in which the student typically starts by asking broad, open-ended questions before drilling down into specifics (Hintikka, 1981; Hakkarainen and Sintonen, 2002). This pedagogical perspective motivates a new way of representing documents. In this paper, we present SQUASH (**S**pecificity-controlled **Qu**estion-**A**n**s**wer **H**ierarchies), a novel and challenging text generation task that converts an input document into a hierarchy of question-answer pairs. Users can click on high-level questions (e.g., "Why did Frodo leave the Fellowship?") to reveal related but more specific questions (e.g., "Who did Frodo leave with?"). Using a question taxonomy loosely based on Lehnert (1978), we classify questions in existing reading comprehension datasets as either GENERAL or SPECIFIC. We then use these labels as input to a pipelined system centered around a conditional neural language model. We extensively evaluate the quality of the generated QA hierarchies through crowdsourced experiments and report strong empirical results.

1 Introduction

Q: *What is this paper about?*
A: We present a novel text generation task which converts an input document into a model-generated hierarchy of question-answer (QA) pairs arranged in a top-down tree structure (Figure 1). Questions at higher levels of the tree are broad and open-ended while questions at lower levels ask about more specific factoids. An entire document has multiple root nodes ("key ideas") that unfold into a forest of question trees. While readers are initially shown only the root nodes of the question trees, they can "browse" the document by clicking on root nodes of interest to reveal more fine-grained related information. We call our task SQUASH (**S**pecificity-controlled **Qu**estion **A**n**s**wer **H**ierarchies).


Massive Attack (band)

On 21 January 2016, the iPhone application "Fantom" was released. The application was developed by a team including Massive Attack's Robert Del Naja and let users hear parts of four new songs by remixing them in real time, using the phone's location, movement, clock, heartbeat, and camera. On 28 January 2016, Massive Attack released a new EP, Ritual Spirit, which includes the four songs released on Fantom.

Q. What was the iPhone application Fantom?
A. The app... let users hear parts of ... real time,

Q. Who created it?
A. ... team including ... Robert Del Naja

Q. What is Ritual Spirit?
A. On ... Attack released a new EP Ritual Spirit

Q. When did they release a song with "Fantom"?
A. On 28 January 2016,

...

On 26 July 2016, Massive Attack previewed three new songs: "Come Near Me", "The Spoils", and "Dear Friend" on Fantom, an iPhone application on which they previously previewed the four songs from the Ritual Spirit EP ... The video for "The Spoils", featuring Cate Blanchett, and directed by Australian director John Hillcoat, ...

Q. What did they do in 2016?
A. On ... 2016 ... three new songs: "Come Near Me", "The Spoils", and "Dear Friend" on Fantom,

Q. Who was in the video?
A. The video ... featuring Cate Blanchett,

Q. Who was the Australian director?
A. John Hillcoat,

...

Figure 1: A subset of the QA hierarchy generated by our SQUASH system that consists of GENERAL and SPECIFIC questions with extractive answers.

Q: *Why represent a document with QA pairs?*[1]
A: Questions and answers (QA) play a critical role in scientific inquiry, information-seeking dialogue and knowledge acquisition (Hintikka, 1981, 1988; Stede and Schlangen, 2004). For example, web users often use QA pairs to manage and share knowledge (Wagner, 2004; Wagner and Bolloju, 2005; Gruber, 2008). Additionally, unstructured lists of "frequently asked questions" (FAQs) are regularly deployed at scale to present information. Industry studies have demonstrated their effectiveness at cutting costs associated with answering customer calls or hiring technical experts (Davenport et al., 1998). Automating the generation of QA pairs can thus be of immense value to companies and web communities.

[1]Our introduction is itself an example of the QA format. Other academic papers such as Henderson et al. (2018) have also used this format to effectively present information.

Q: *Why add hierarchical structure to QA pairs?*
A: While unstructured FAQs are useful, pedagogical applications benefit from additional hierarchical organization. Hakkarainen and Sintonen (2002) show that students learn concepts effectively by first asking general, explanation-seeking questions before drilling down into more specific questions. More generally, hierarchies break up content into smaller, more digestable chunks. User studies demonstrate a strong preference for hierarchies in document summarization (Buyukkokten et al., 2001; Christensen et al., 2014) since they help readers easily identify and explore key topics (Zhang et al., 2017).

Q: *How do we build systems for* SQUASH*?*
A: We leverage the abundance of reading comprehension QA datasets to train a pipelined system for SQUASH. One major challenge is the lack of labeled hierarchical structure within existing QA datasets; we tackle this issue in Section 2 by using the question taxonomy of Lehnert (1978) to classify questions in these datasets as either GENERAL or SPECIFIC. We then condition a neural question generation system on these two classes, which enables us to generate both types of questions from a paragraph. We filter and structure these outputs using the techniques described in Section 3.

Q: *How do we evaluate our* SQUASH *pipeline?*
A: Our crowdsourced evaluation (Section 4) focuses on fundamental properties of the generated output such as QA quality, relevance, and hierarchical correctness. Our work is a first step towards integrating QA generation into document understanding; as such, we do not directly evaluate how *useful* SQUASH output is for downstream pedagogical applications. Instead, a detailed qualitative analysis (Section 5) identifies challenges that need to be addressed before SQUASH can be deployed to real users.

Q: *What are our main contributions?*
A1: A method to classify questions according to their specificity based on Lehnert (1978).
A2: A model controlling specificity of generated questions, unlike prior work on QA generation.
A3: A novel text generation task (SQUASH), which converts documents into specificity-based hierarchies of QA pairs.
A4: A pipelined system to tackle SQUASH along with crowdsourced methods to evaluate it.

Q: *How can the community build on this work?*
A: We have released our codebase, crowdsourcing templates for evaluation, and a live demonstration of our system at `http://squash.cs.umass.edu/`. Additionally, we outline guidelines for future work in Section 7.

2 Obtaining training data for SQUASH

The proliferation of reading comprehension datasets like SQuAD (Rajpurkar et al., 2016, 2018) has enabled state-of-the-art neural question generation systems (Du et al., 2017; Kim et al., 2018). However, these systems are trained for *individual* question generation, while the goal of SQUASH is to produce a general-to-specific hierarchy of QA pairs. Recently-released conversational QA datasets like QuAC (Choi et al., 2018) and CoQA (Reddy et al., 2018) contain a *sequential* arrangement of QA pairs, but question specificity is not explicitly marked.[2] Motivated by the lack of hierarchical QA datasets, we automatically classify questions in SQuAD, QuAC and CoQA according to their specificity using a combination of rule-based and automatic approaches.

2.1 Rules for specificity classification

What makes one question more specific than another? Our scheme for classifying question specificity maps each of the 13 conceptual question categories defined by Lehnert (1978) to three coarser labels: GENERAL, SPECIFIC, or YES-NO.[3] As a result of this mapping, SPECIFIC questions usually ask for low-level information (e.g., entities or numerics), while GENERAL questions ask for broader overviews (e.g., "what happened in 1999?") or causal information (e.g, "why did..."). Many question categories can be reliably identified using simple templates and rules; A complete list is provided in Table 1.[4]

Classifying questions not covered by templates: If a question does not satisfy any template or rule, how do we assign it a label? We manage to clas-

[2]"Teachers" in the QuAC set-up can encourage "students" to ask a follow-up question, but we cannot use these annotations to infer a hierarchy because students are not required to actually follow their teachers' directions.

[3]We add a third category for YES-NO questions as they are difficult to classify as either GENERAL or SPECIFIC.

[4]Questions in Lehnert (1978) were classified using a conceptual dependency parser (Schank, 1972). We could not find a modern implementation of this parser and thus decided to use a rule-based approach that relies on spaCy 2.0 (Honnibal and Montani, 2017) for all preprocessing.

Conceptual class	**Specificity**	**Question asks for...**	**Sample templates**
Causal Antecendent, Goal Oriented, Enablement, Causal Consequent, Expectational	GENERAL	the reason for occurrence of an event and the consequences of it	*Why ..., What happened after / before ..., What was the cause / reason / purpose ..., What led to ...*
Instrumental	GENERAL	a procedure / mechanism	*How* question with VERB parent for *How* in dependency tree
Judgemental	GENERAL	a listener's opinion	Words like *you*, *your* present
Concept Completion, Feature Specification	GENERAL *or* SPECIFIC	fill-in-the-blank information	*Where / When / Who ...* ("SPECIFIC" templates)
Quantification	SPECIFIC	an amount	*How many / long ...*
Verification, Disjunctive	YES-NO	Yes-No answers	first word is VERB
Request	N/A	an act to be performed	(absent in datasets)

Table 1: The 13 conceptual categories of Lehnert (1978) and some templates to identify them and their specificity.

sify roughly half of all questions with our templates and rules (Table A1); for the remaining half, we resort to a data-driven approach. First, we manually label 1000 questions in QuAC[5] using our specificity labels. This annotated data is then fed to a single-layer CNN binary classifier (Kim, 2014) using ELMo contextualized embeddings (Peters et al., 2018).[6] On a 85%-15% train-validation split, we achieve a high classification accuracy of 91%. The classifier also transfers to other datasets: on 100 manually labeled CoQA questions, we achieve a classification accuracy of 80%. To obtain our final dataset (Table 2), we run our rule-based approach on all questions in SQuAD 2.0, QuAC, and CoQA and apply our classifier to label questions that were not covered by the rules. We further evaluate the specificity of the questions generated by our final system using a crowdsourced study in Section 4.3.

Dataset	Size	GENERAL	SPECIFIC	YES-NO
SQuAD	86.8k	28.2%	69.7%	2.1%
QuAC	65.2k	34.9%	33.5%	31.6%
CoQA	105.6k	23.6%	54.9%	21.5%
All	257.6k	28.0%	54.5%	17.5%

Table 2: Distribution of classes in the final datasets. We add some analysis on this distribution in Appendix A.

3 A pipeline for SQUASHing documents

To SQUASH documents, we build a pipelined system (Figure 2) that takes a single paragraph as input and produces a hierarchy of QA pairs as output; for multi-paragraph documents, we SQUASH each paragraph independently of the rest. At a high level, the pipeline consists of five steps: (1) answer span selection, (2) question generation conditioned on answer spans and specificity labels, (3) *extractively* answering generated questions, (4) filtering out bad QA pairs, and (5) structuring the remaining pairs into a GENERAL-to-SPECIFIC hierarchy. The remainder of this section describes each step in more detail.

3.1 Answer span selection

Our pipeline begins by selecting an answer span from which to generate a question. To train the system, we can use ground-truth answer spans from our labeled datasets, but at test time how do we select answer spans? Our solution is to consider all individual sentences in the input paragraph as potential answer spans (to generate GENERAL and SPECIFIC questions), along with all entities and numerics (for just SPECIFIC questions). We did not use data-driven sequence tagging approaches like previous work (Du and Cardie, 2017, 2018), since our preliminary experiments with such approaches yielded poor results on QuAC.[7] More details are provided in Appendix C.

3.2 Conditional question generation

Given a paragraph, answer span, and desired specificity label, we train a neural encoder-decoder model on all three reading comprehension datasets (SQuAD, QuAC and CoQA) to generate an appropriate question.

Data preprocessing: At training time, we use the ground-truth answer spans from these datasets

[5]We use QuAC because its design encourages a higher percentage of GENERAL questions than other datasets, as the question-asker was unable to read the document to formulate more specific questions.

[6]Implemented in AllenNLP (Gardner et al., 2018).

[7]We hypothesize that answer span identification on QuAC is difficult because the task design encouraged "teachers" to provide more information than just the minimal answer span.

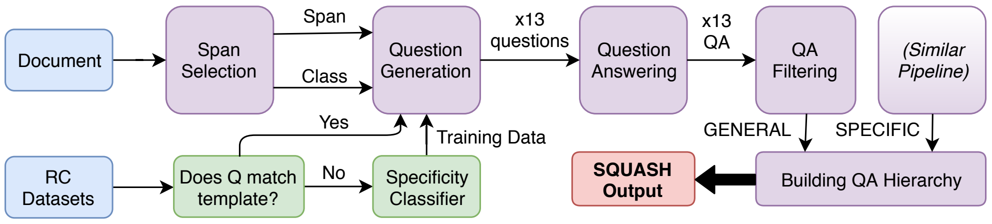


Figure 2: An overview of the process by which we generate a pair of GENERAL-SPECIFIC questions, which consists of feeding input data (*"RC"* is Reading Comprehension) through various modules, including a question classifier and a multi-stage pipeline for question generation, answering, and filtering.

as input to the question generator. To improve the quality of SPECIFIC questions generated from sentence spans, we use the extractive *evidence* spans for CoQA instances (Reddy et al., 2018) instead of the shorter, partially abstractive answer spans (Yatskar, 2019). In all datasets, we remove unanswerable questions and questions whose answers span multiple paragraphs. A few very generic questions (e.g. "what happened in this article?") were manually identified removed from the training dataset. Some other questions (e.g., "where was he born?") are duplicated many times in the dataset; we downsample such questions to a maximum limit of 10. Finally, we preprocess both paragraphs and questions using byte-pair encoding (Sennrich et al., 2016).

Architecture details: We use a two-layer biLSTM encoder and a single-layer LSTM (Hochreiter and Schmidhuber, 1997) decoder with soft attention (Bahdanau et al., 2015) to generate questions, similar to Du et al. (2017). Our architecture is augmented with a copy mechanism (See et al., 2017) over the encoded paragraph representations. Answer spans are marked with <SOA> and <EOA> tokens in the paragraph, and representations for tokens within the answer span are attended to by a separate attention head. We condition the decoder on the specificity class (GENERAL, SPECIFIC and YES-NO)[8] by concatenating an embedding for the ground-truth class to the input of each time step. We implement models in PyTorch `v0.4` (Paszke et al., 2017), and the best-performing model achieves a perplexity of 11.1 on the validation set. Other hyperparameters details are provided in Appendix B.

Test time usage: At test time, the question generation module is supplied with answer spans and class labels as described in Section 3.1. To promote diversity, we over-generate prospective candidates (Heilman and Smith, 2010) for every answer span and later prune them. Specifically, we use beam search with a beam size of 3 to generate three highly-probable question candidates. As these candidates are often generic, we additionally use *top-k random sampling* (Fan et al., 2018) with $k = 10$, a recently-proposed diversity-promoting decoding algorithm, to generate ten more question candidates per answer span. Hence, for every answer span we generate **13** question candidates. We discuss issues with using just standard beam search for question generation in Section 5.1.

[8]While we do not use YES-NO questions at test time, we keep this class to avoid losing a significant proportion of training data.

3.3 Answering generated questions

While we condition our question generation model on pre-selected answer spans, the generated questions may not always correspond to these input spans. Sometimes, the generated questions are either unanswerable or answered by a different span in the paragraph. By running a pretrained QA model over the generated questions, we can detect questions whose answers do not match their original input spans and filter them out. The predicted answer for many questions has partial overlap with the original answer span; in these cases, we display the predicted answer span during evaluation, as a qualitative inspection shows that the predicted answer is more often closer to the correct answer. For all of our experiments, we use the AllenNLP implementation of the BiDAF++ question answering model of Choi et al. (2018) trained on QuAC with no dialog context.

3.4 Question filtering

After over-generating candidate questions from a single answer span, we use simple heuristics to filter out low-quality QA pairs. We remove generic and duplicate question candidates[9] and pass the remaining QA pairs through the multi-stage question filtering process described below.

Irrelevant or repeated entities: *Top-k random sampling* often generates irrelevant questions; we reduce their incidence by removing any candidates that contain nouns or entities unspecified in the passage. As with other neural text generation systems (Holtzman et al., 2018), we commonly observe repetition in the generated questions and deal with this phenomenon by removing candidates with repeated nouns or entities.

Unanswerable or low answer overlap: We remove all candidates marked as "unanswerable" by the question answering model, which prunes 39.3% of non-duplicate question candidates. These candidates are generally grammatically correct but considered irrelevant to the original paragraph by the question answering model. Next, we compute the overlap between original and predicted answer span by computing word-level precision and recall (Rajpurkar et al., 2016). For GENERAL questions generated from sentence spans, we attempt to maximize recall by setting a minimum recall threshold of 0.3.[10] Similarly, we maximize recall for SPECIFIC questions generated from named entities with a minimum recall constraint of 0.8. Finally, for SPECIFIC questions generated from sentence spans, we set a minimum *precision* threshold of 1.0, which filters out questions whose answers are not completely present in the ground-truth sentence.

Low generation probability: If multiple candidates remain after applying the above filtering criteria, we select the most probable candidate for each answer span. SPECIFIC questions generated from sentences are an exception to this rule: for these questions, we select the ten most probable candidates, as there might be multiple question-worthy bits of information in a single sentence. If no candidates remain, in some cases[11] we use a fallback mechanism that sequentially ignores filters to retain more candidates.

[9]Running *Top-k random sampling* multiple times can produce duplicate candidates, including those already in the top beams.

[10]Minimum thresholds were qualitatively chosen based on the specificity type.

[11]For example, if no valid GENERAL questions for the entire paragraph are generated.

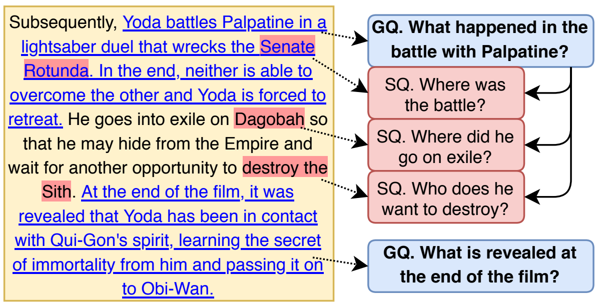

Subsequently, Yoda battles Palpatine in a lightsaber duel that wrecks the Senate Rotunda. In the end, neither is able to overcome the other and Yoda is forced to retreat. He goes into exile on Dagobah so that he may hide from the Empire and wait for another opportunity to destroy the Sith. At the end of the film, it was revealed that Yoda has been in contact with Qui-Gon's spirit, learning the secret of immortality from him and passing it on to Obi-Wan.
GQ. What happened in the battle with Palpatine?
SQ. Where was the battle?
SQ. Where did he go on exile?
SQ. Who does he want to destroy?
GQ. What is revealed at the end of the film?

Figure 3: Procedure used to form a QA hierarchy. The predicted answers for GQs (GENERAL questions), are underlined in blue. The predicted answers for SQs (SPECIFIC questions) are highlighted in red.

3.5 Forming a QA hierarchy

The output of the filtering module is an unstructured list of GENERAL and SPECIFIC QA pairs generated from a single paragraph. Figure 3 shows how we group these questions into a meaningful hierarchy. First, we choose a parent for each SPECIFIC question by maximizing the overlap (word-level precision) of its predicted answer with the predicted answer for every GENERAL question. If a SPECIFIC question's answer does not overlap with any GENERAL question's answer (e.g., *"Dagobah"* and *"destroy the Sith"*) we map it to the closest GENERAL question whose answer occurs *before* the SPECIFIC question's answer (*"What happened in the battle ...?"*).[12]

4 Evaluation

We evaluate our SQUASH pipeline on documents from the QuAC development set using a variety of crowdsourced[13] experiments. Concretely, we evaluate the quality and relevance of individual questions, the relationship between generated questions and predicted answers, and the structural properties of the QA hierarchy. We emphasize that our experiments examine only the *quality* of a SQUASHed document, not its actual *usefulness* to downstream users. Evaluating usefulness (e.g., measuring if SQUASH is more helpful than the input document) requires systematic and targeted human studies (Buyukkokten et al., 2001) that are beyond the scope of this work.

[12]This heuristic is justified because users read GENERAL questions *before* SPECIFIC ones in our interface.

[13]All our crowdsourced experiments were conducted on the Figure Eight platform with three annotators per example (scores calculated by counting examples with two or more correct judgments). We hired annotators from predominantly English-speaking countries with a rating of at least Level 2, and we paid them between 3 and 4 cents per judgment.

Experiment	Generated		Gold	
	Score	Fleiss κ	Score	Fleiss κ
Is this question well-formed?	85.8%	0.65	93.3%	0.54
Is this question relevant?	78.7%	0.36	83.3%	0.41
(among *well-formed*)	81.1%	0.39	83.3%	0.40
Does the span *partially* contain the answer?	85.3%	0.45	81.1%	0.43
(among *well-formed*)	87.6%	0.48	82.1%	0.42
(among *well-formed and relevant*)	94.9%	0.41	92.9%	0.44
Does the span *completely* contain the answer?	74.1%	0.36	70.0%	0.37
(among *well-formed*)	76.9%	0.36	70.2%	0.39
(among *well-formed and relevant*)	85.4%	0.30	80.0%	0.42

Table 3: Human evaluations demonstrate the high individual QA quality of our pipeline's outputs. All interannotator agreement scores (Fleiss κ) show "fair" to "substantial" agreement (Landis and Koch, 1977).

4.1 Individual question quality and relevance

Our first evaluation measures whether questions generated by our system are *well-formed* (i.e., grammatical and pragmatic). We ask crowd workers whether or not a given question is both grammatical and meaningful.[14] For this evaluation, we acquire judgments for 200 generated QA pairs and 100 gold QA pairs[15] from the QuAC validation set (with an equal split between GENERAL and SPECIFIC questions). The first row of Table 3 shows that 85.8% of generated questions satisfy this criterion with a high agreement across workers.

Question relevance: How many generated questions are actually relevant to the input paragraph? While the percentage of *unanswerable* questions that were generated offers some insight into this question, we removed all of them during the filtering pipeline (Section 3.4). Hence, we display an input paragraph and generated question to crowd workers (using the same data as the previous well-formedness evaluation) and ask whether or not the paragraph contains the answer to the question. The second row of Table 3 shows that 78.7% of our questions are relevant to the paragraph, compared to 83.3% of gold questions.

4.2 Individual answer validity

Is the predicted answer actually a valid answer to the generated question? In our filtering process, we automatically measured answer overlap between the input answer span and the predicted answer span and used the results to remove low-overlap QA pairs. To evaluate answer recall after filtering, we perform a crowdsourced evaluation on the same 300 QA pairs as above by asking crowdworkers whether or not a predicted answer span *contains* the answer to the question. We also experiment with a more relaxed variant (*partially contains* instead of *completely contains*) and report results for both task designs in the third and fourth rows of Table 3. Over 85% of predicted spans partially contain the answer to the generated question, and this number increases if we consider only questions that were previously labeled as *well-formed* and *relevant*. The lower gold performance is due to the contextual nature of the gold QA pairs in QuAC, which causes some questions to be meaningless in isolation (e.g. *"What did she do next?"* has unresolvable coreferences).

Experiment	Score	Fleiss κ
Which question type asks for more information?	89.5%	0.57
Which SPECIFIC question is closer to GENERAL QA?		
different paragraph	77.0%	0.47
same paragraph	64.0%	0.30

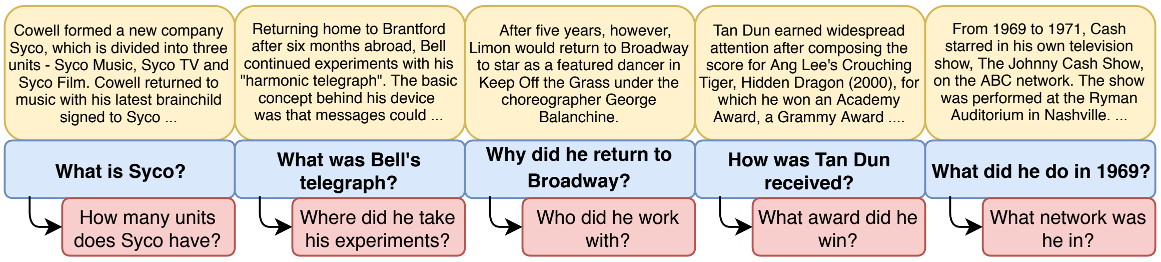
Table 4: Human evaluation of the structural correctness of our system. The labels *"different / same paragraph"* refer to the location of the *intruder* question. The results show the accuracy of specificity and hierarchies.

4.3 Structural correctness

To examine the hierachical structure of SQUASHed documents, we conduct three experiments.

[14] As "meaningful" is potentially a confusing term for crowd workers, we ran another experiment asking only for grammatical correctness and achieved very similar results.

[15] Results on this experiment were computed after removing 3 duplicate generated questions and 10 duplicate gold questions.


Cowell formed a new company Syco, which is divided into three units - Syco Music, Syco TV and Syco Film. Cowell returned to music with his latest brainchild signed to Syco ...

What is Syco?

How many units does Syco have?

Returning home to Brantford after six months abroad, Bell continued experiments with his "harmonic telegraph". The basic concept behind his device was that messages could ...

What was Bell's telegraph?

Where did he take his experiments?

After five years, however, Limon would return to Broadway to star as a featured dancer in Keep Off the Grass under the choreographer George Balanchine.

Why did he return to Broadway?

Who did he work with?

Tan Dun earned widespread attention after composing the score for Ang Lee's Crouching Tiger, Hidden Dragon (2000), for which he won an Academy Award, a Grammy Award ....

How was Tan Dun received?

What award did he win?

From 1969 to 1971, Cash starred in his own television show, The Johnny Cash Show, on the ABC network. The show was performed at the Ryman Auditorium in Nashville. ...

What did he do in 1969?

What network was he in?

Figure 4: SQUASH question hierarchies generated by our system with reference snippets. Questions in the hierarchy are of the correct specificity class (i.e., GENERAL, SPECIFIC).

How faithful are output questions to input specificity? First, we investigate whether our model is actually generating questions with the correct specificity label. We run our specificity classifier (Section 2) over 400 randomly sampled questions (50% GENERAL, 50% SPECIFIC) and obtain a high classification accuracy of 91%.[16] This automatic evaluation suggests the model is capable of generating different types of questions.

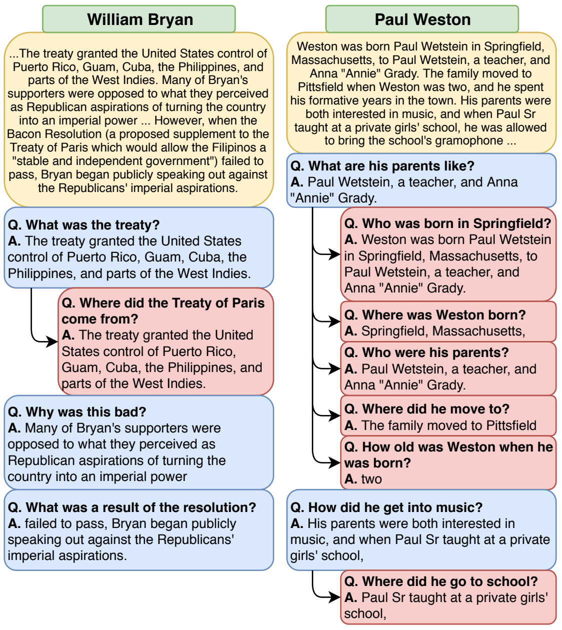
Are GENERAL questions more representative of a paragraph than SPECIFIC questions? To see if GENERAL questions really do provide more high-level information, we sample 200 GENERAL-SPECIFIC question pairs[17] grouped together as described in Section 3.5. For each pair of questions (without showing answers), we ask crowd workers to choose the question which, if answered, would give them more information about the paragraph. As shown in Table 4, in 89.5% instances the GENERAL question is preferred over the SPECIFIC one, which confirms the strength of our specificity-controlled question generation system.[18]

How related are SPECIFIC questions to their parent GENERAL question? Finally, we investigate the effectiveness of our question grouping strategy, which bins multiple SPECIFIC QA pairs under a single GENERAL QA pair. We show crowd workers a reference GENERAL QA pair and ask them to choose the most related SPECIFIC question given two choices, one of which is the system's output and the other an *intruder* question.

[16]Accuracy computed after removing 19 duplicates.

[17]We avoid gold-standard control experiments for structural correctness tests since questions in the QuAC dataset were not generated with a hierarchical structure in mind. Pilot studies using our question grouping module on gold data led to sparse hierarchical structures which were not favored by our crowd workers.

[18]We also ran a pilot study asking workers *"Which question has a longer answer?"* and observed a higher preference of 98.6% for GENERAL questions.


William Bryan

...The treaty granted the United States control of Puerto Rico, Guam, Cuba, the Philippines, and parts of the West Indies. Many of Bryan's supporters were opposed to what they perceived as Republican aspirations of turning the country into an imperial power ... However, when the Bacon Resolution (a proposed supplement to the Treaty of Paris which would allow the Filipinos a "stable and independent government") failed to pass, Bryan began publicly speaking out against the Republicans' imperial aspirations.

Q. What was the treaty?
A. The treaty granted the United States control of Puerto Rico, Guam, Cuba, the Philippines, and parts of the West Indies.

Q. Where did the Treaty of Paris come from?
A. The treaty granted the United States control of Puerto Rico, Guam, Cuba, the Philippines, and parts of the West Indies.

Q. Why was this bad?
A. Many of Bryan's supporters were opposed to what they perceived as Republican aspirations of turning the country into an imperial power

Q. What was a result of the resolution?
A. failed to pass, Bryan began publicly speaking out against the Republicans' imperial aspirations.

Paul Weston

Weston was born Paul Wetstein in Springfield, Massachusetts, to Paul Wetstein, a teacher, and Anna "Annie" Grady. The family moved to Pittsfield when Weston was two, and he spent his formative years in the town. His parents were both interested in music, and when Paul Sr taught at a private girls' school, he was allowed to bring the school's gramophone ...

Q. What are his parents like?
A. Paul Wetstein, a teacher, and Anna "Annie" Grady.

Q. Who was born in Springfield?
A. Weston was born Paul Wetstein in Springfield, Massachusetts, to Paul Wetstein, a teacher, and Anna "Annie" Grady.

Q. Where was Weston born?
A. Springfield, Massachusetts,

Q. Who were his parents?
A. Paul Wetstein, a teacher, and Anna "Annie" Grady.

Q. Where did he move to?
A. The family moved to Pittsfield

Q. How old was Weston when he was born?
A. two

Q. How did he get into music?
A. His parents were both interested in music, and when Paul Sr taught at a private girls' school,

Q. Where did he go to school?
A. Paul Sr taught at a private girls' school,

Figure 5: Two SQUASH outputs generated by our system. The William Bryan example has interesting GENERAL questions. The Paul Weston example showcases several mistakes our model makes.

We randomly select intruder SPECIFIC questions from either a different paragraph within the same document or a different group within the same paragraph. As shown in Table 4, crowd workers prefer the system's generated SPECIFIC question with higher than random chance (50%) regardless of where the intruder comes from. As expected, the preference and agreement is higher when intruder questions come from different paragraphs, since groups within the same paragraph often contain related information (Section 5.2).

5 Qualitative Analysis

In this section we analyze outputs (Figure 4, Figure 5) of our pipeline and identify its strengths and weaknesses. We additionally provide more examples in the appendix (Figure A1).

"In ***1942****, Dodds enlisted in the US army and served as an anti aircraft gunner during World War II."*	
B	In what year did the US army take place? In what year did the US army take over? In what year did the US army take place in the US?
T	What year was he enlisted? When did he go to war? When did he play as anti aircraft?

Table 5: Beam Search (B) vs Top-*k* sampling (T) for SPECIFIC question generation. Top-*k* candidates tend to be more diverse.

5.1 What is our pipeline good at?

Meaningful hierarchies: Our method of grouping the generated questions (Section 3.5) produces hierarchies that clearly distinguish between GENERAL and SPECIFIC questions; Figure 4 contains some hierarchies that support the positive results of our crowdsourced evaluation.

***Top-k* sampling:** Similar to prior work (Fan et al., 2018; Holtzman et al., 2019), we notice that beam search often produces generic or repetitive beams (Table 5). Even though the *top-k* scheme always produces lower-probable questions than beam search, our filtering system prefers a *top-k* question **49.5%** of the time.

5.2 What kind of mistakes does it make?

We describe the various types of errors our model makes in this section, using the Paul Weston SQUASH output in Figure 5 as a running example. Additionally, we list some modeling approaches we tried that did *not* work in Appendix C.

Reliance on a flawed answering system: Our pipeline's output is tied to the quality of the pretrained answering module, which both filters out questions and produces final answers. QuAC has long answer spans (Choi et al., 2018) that cause low-precision predictions with extra information (e.g., *"Who was born in Springfield?"*). Additionally, the answering module occasionally swaps two named entities present in the paragraph.[19]

Redundant information and lack of discourse: In our system, each QA pair is generated independently of all the others. Hence, our outputs lack an inter-question discourse structure. Our system often produces a pair of redundant SPECIFIC questions where the text of one question answers the other (e.g., *"Who was born in Springfield?"* vs. *"Where was Weston born?"*). These errors can likely be corrected by conditioning the generation module on previously-produced questions (or additional filtering); we leave this to future work.

Lack of world knowledge: Our models lack commonsense knowledge (*"How old was Weston when he was born?"*) and can misinterpret polysemous words. Integrating pretrained contextualized embeddings (Peters et al., 2018) into our pipeline is one potential solution.

Multiple GENERAL QA per paragraph: Our system often produces more than one tree per paragraph, which is undesirable for short, focused paragraphs with a single topic sentence. To improve the user experience, it might be ideal to restrict the number of GENERAL questions we show per paragraph. While we found it difficult to generate GENERAL questions representative of entire paragraphs (Appendix C), a potential solution could involve identifying and generating questions from topic sentences.

Coreferences in GENERAL questions: Many generated GENERAL questions contain coreferences due to contextual nature of the QuAC and CoQA training data (*"How did* ***he*** *get into music?"*). Potential solutions could involve either constrained decoding to avoid beams with anaphoric expressions or using the CorefNQG model of Du and Cardie (2018).

5.3 Which models did not work?

We present modelling approaches which did not work in Appendix C. This includes, i) end-to-end modelling to generate sequences of questions using QuAC, ii) span selection NER system, iii) generation of GENERAL questions representative of entire paragraphs, iv) answering system trained on the combination of QuAC, CoQA and SQuAD.

6 Related Work

Our work on SQUASH is related to research in three broad areas: question generation, information retrieval and summarization.

Question Generation: Our work builds upon neural question generation systems (Du et al., 2017; Du and Cardie, 2018). Our work conditions generation on specificity, similar to difficulty-conditioned question generation (Gao et al., 2018). QA pair generation has previously been used for

[19]For instance in the sentence *"The Carpenter siblings were born in New Haven, to Harold B. and Agnes R."* the model *incorrectly* answers the question *"Who was born in New Haven?"* as *"Harold B. and Agnes R."*

dataset creation (Serban et al., 2016; Du and Cardie, 2018). Joint modeling of question generation and answering has improved the performance of individual components (Tang et al., 2017; Wang et al., 2017; Sachan and Xing, 2018) and enabled visual dialog generation (Jain et al., 2018).

Information Retrieval: Our hierarchies are related to interactive retrieval setting (Hardtke et al., 2009; Brandt et al., 2011) where similar webpages are grouped together. SQUASH is also related to exploratory (Marchionini, 2006) and faceted search (Yee et al., 2003).

Summarization: Our work is related to *query-focused* summarization (Dang, 2005; Baumel et al., 2018) which conditions an output summary on an input query. Hierarchies have also been applied to summarization (Christensen et al., 2014; Zhang et al., 2017; Tauchmann et al., 2018).

7 Future Work

While Section 5.2 focused on shortcomings in our modeling process and steps to fix them, this section focuses on broader guidelines for future work involving the SQUASH format and its associated text generation task.

Evaluation of the SQUASH format: As discussed in Section 1, previous research shows support for the usefulness of hierarchies and QA in pedagogical applications. We did not directly evaluate this claim in the context of SQUASH, focusing instead on evaluating the quality of QA pairs and their hierarchies. Moving forward, careful user studies are needed to evaluate the efficacy of the SQUASH format in pedagogical applications, which might be heavily domain-dependent; for example, a QA hierarchy for a research paper is likely to be more useful to an end user than a QA hierarchy for an online blog. An important caveat is the imperfection of modern text generation systems, which might cause users to prefer the original human-written document over a generated SQUASH output. One possible solution is a three-way comparison between the original document, a human-written SQUASHed document, and a system-generated output. For fair comparison, care should be taken to prevent experimenter bias while crowdsourcing QA hierarchies (e.g., by maintaining similar text complexity in the two human-written formats).

Collection of a SQUASH dataset: Besides measuring the usefulness of the QA hierarchies, a large dedicated dataset can help to facilitate end-to-end modeling. While asking human annotators to write full SQUASHed documents will be expensive, a more practical option is to ask them to pair GENERAL and SPECIFIC questions in our dataset to form meaningful hierarchies and write extra questions whenever no such pair exists.

QA budget and deeper specificity hierarchies: In our work, we generate questions for every sentence and filter bad questions with fixed thresholds. An alternative formulation is an adaptive model dependent on a user-specified QA budget, akin to "target length" in summarization systems, which would allow end users to balance coverage and brevity themselves. A related modification is increasing the depth of the hierarchies. While two-level QA trees are likely sufficient for documents structured into short and focused paragraphs, deeper hierarchies can be useful for long unstructured chunks of text. Users can control this property via a "maximum children per QA node" hyperparameter, which along with the QA budget will determine the final depth of the hierarchy.

8 Conclusion

We propose SQUASH, a novel text generation task which converts a document into a hierarchy of QA pairs. We present and evaluate a system which leverages existing reading comprehension datasets to attempt solving this task. We believe SQUASH is a challenging text generation task and we hope the community finds it useful to benchmark systems built for document understanding, question generation and question answering. Additionally, we hope that our specificity-labeled reading comprehension dataset is useful in other applications such as 1) finer control over question generation systems used in education applications, curiosity-driven chatbots and healthcare (Du et al., 2017).

Acknowledgements

We thank the anonymous reviewers for their insightful comments. In addition, we thank Nader Akoury, Ari Kobren, Tu Vu and the other members of the UMass NLP group for helpful comments on earlier drafts of the paper and suggestions on the paper's presentation. This work was supported in part by research awards from the Allen Institute for Artificial Intelligence and Adobe Research.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. International Conference on Learning Representations (ICLR)*.

Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *arXiv preprint arXiv:1801.07704*.

Christina Brandt, Thorsten Joachims, Yisong Yue, and Jacob Bank. 2011. Dynamic ranked retrieval. In *Proceedings of the fourth ACM international conference on Web search and data mining*.

Orkut Buyukkokten, Hector Garcia-Molina, and Andreas Paepcke. 2001. Seeing the whole in parts: text summarization for web browsing on handheld devices. In *Proceedings of the 10th international conference on World Wide Web*.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Janara Christensen, Stephen Soderland, Gagan Bansal, et al. 2014. Hierarchical summarization: Scaling up multi-document summarization. In *Proc. Association for Computational Linguistics (ACL)*.

Hoa Trang Dang. 2005. Overview of duc 2005. In *Document Understanding Conferences*.

Thomas H Davenport, David W De Long, and Michael C Beers. 1998. Successful knowledge management projects. *Sloan management review*, 39(2):43–57.

Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from wikipedia. In *Proc. Association for Computational Linguistics (ACL)*.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proc. Association for Computational Linguistics (ACL)*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proc. Association for Computational Linguistics (ACL)*.

Yifan Gao, Jianan Wang, Lidong Bing, Irwin King, and Michael R Lyu. 2018. Difficulty controllable question generation for reading comprehension. *arXiv preprint arXiv:1807.03586*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*.

Tom Gruber. 2008. Collective knowledge systems: Where the social web meets the semantic web. *Web semantics: science, services and agents on the World Wide Web*, 6(1).

Kai Hakkarainen and Matti Sintonen. 2002. The interrogative model of inquiry and computer-supported collaborative learning. *Science & Education*, 11(1):25–43.

David Hardtke, Mike Wertheim, and Mark Cramer. 2009. Demonstration of improved search result relevancy using real-time implicit relevance feedback. *Understanding the User-Logging and Interpreting User Interactions in Information Search and Retrieval (UIIR-2009)*.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Proc. Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT)*.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In *Proc. Association for the Advancement of Artificial Intelligence (AAAI)*.

Jaakko Hintikka. 1981. The logic of information-seeking dialogues: A model. *Werner Becker and Wilhelm K. Essler Konzepte der Dialektik*, pages 212–231.

Jaakko Hintikka. 1988. What is the logic of experimental inquiry? *Synthese*, 74(2):173–190.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proc. Association for Computational Linguistics (ACL)*.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Unnat Jain, Svetlana Lazebnik, and Alexander G Schwing. 2018. Two can play this game: visual dialog with discriminative question generation and answering. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR)*.

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2018. Improving neural question generation using answer separation. *arXiv preprint arXiv:1809.02393*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations (ICLR)*.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Wendy G Lehnert. 1978. *The process of question answering: A computer simulation of cognition*, volume 978. Lawrence Erlbaum Hillsdale, NJ.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Gary Marchionini. 2006. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS 2017 Autodiff Workshop*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT)*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proc. Association for Computational Linguistics (ACL)*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing*, pages 2383–2392.

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

Mrinmaya Sachan and Eric Xing. 2018. Self-training for jointly learning to ask and answer questions. In *Proc. Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT)*.

Roger C Schank. 1972. Conceptual dependency: A theory of natural language understanding. *Cognitive psychology*, 3(4):552–631.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proc. Association for Computational Linguistics (ACL)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. Association for Computational Linguistics (ACL)*.

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 *Proc. Association for Computational Linguistics (ACL)*.

Manfred Stede and David Schlangen. 2004. Information-seeking chat: Dialogues driven by topic-structure. In *Proceedings of Catalog (the 8th workshop on the semantics and pragmatics of dialogue; SemDial04)*.

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

Christopher Tauchmann, Thomas Arnold, Andreas Hanselowski, Christian M Meyer, and Margot Mieskes. 2018. Beyond generic summarization: A multi-faceted hierarchical summarization corpus of large heterogeneous data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*.

Christian Wagner. 2004. Wiki: A technology for conversational knowledge management and group collaboration. *Communications of the association for information systems*, 13(1):19.

Christian Wagner and Narasimha Bolloju. 2005. Supporting knowledge management in organizations with conversational technologies: Discussion forums, weblogs, and wikis. *Journal of Database Management*, 16(2).

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

Mark Yatskar. 2019. A qualitative comparison of coqa, squad 2.0 and quac. *Proc. Human Language Technology/Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*.

Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. 2003. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*.

Amy X. Zhang, Lea Verou, and David Karger. 2017. Wikum: Bridging discussion forums and wikis using recursive summarization. In *Conference on Computer Supported Cooperative Work and Social Computing (CSCW)*.

Appendix

A Question Classification Details

Confirming our intuition, Table 2 shows us that QuAC has the highest percentage of GENERAL questions. On the other hand CoQA and SQuAD, which allowed the question-asker to look at the passage, are dominated by SPECIFIC questions. These findings are consistent with a comparison across the three datasets in Yatskar (2019). Interestingly, the average answer length for SPECIFIC questions in QuAC is 12 tokens, compared to 17 tokens for GENERAL questions. We provide the exact distribution of rule-labeled, hand-labeled and classifier-labeled questions in Table A1.

B Hyperparameters for Question Generation

Our question generation system consists of a two layer bidirectional LSTM encoder and a unidirectional LSTM decoder respectively. The LSTM hidden unit size in each direction and token embedding size is each set to 512. The class specificity embeddings size is 16. Embeddings are shared between the paragraph encoder and question decoder. All attention computations use a bilinear product (Luong et al., 2015). A dropout of 0.5 is used between LSTM layers. Models are trained using Adam (Kingma and Ba, 2014) with a learning rate of 10^{-3}, with a gradient clipping of 5.0 and minibatch size 32. Early stopping on validation perplexity is used to choose the best question generation model.

C What did not work?

End-to-End Sequential Generation. We experimented with an end-to-end neural model which generated a sequence of questions given a sequence of answer spans. As training data, we leveraged the sequence IDs and follow-up information in the QuAC dataset, *without* specificity labels. We noticed that during decoding the model rarely attended over the history and often produced questions irrelevant to the context. A potential future direction would involve using the specificity labels for an end-to-end model.

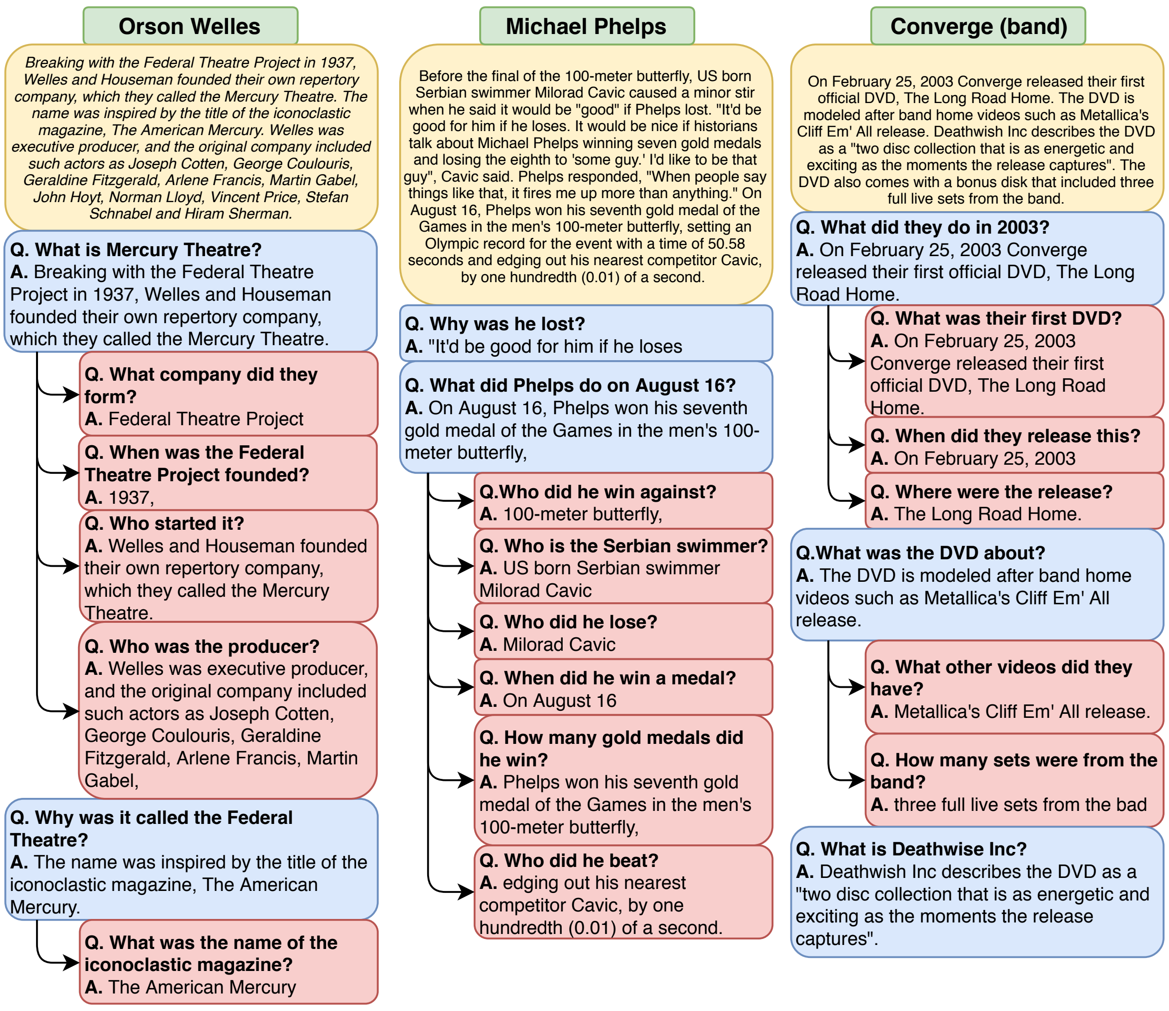
Span Selection NER system. As discussed in Section 3.1 and Du and Cardie (2017), we could frame answer span selection as a sequence labelling problem. We experimented with the NER system in AllenNLP (with ELMo embeddings) on the QuAC dataset, after the ground truth answer spans marked with BIO tags, after overlapping answers were merged together. We recorded low F1 scores of 33.3 and 15.6 on sentence-level and paragraph-level input respectively.

Paragraph-level question generation. Our question generation model rarely generated GENERAL questions representative of the entire paragraph, even when we fed the entire paragraph as the answer span. We noticed that most GENERAL questions in our dataset were answered by one or two sentences in the paragraph.

Answering system trained on all datasets. Recently, Yatskar (2019) reported small improvements on the QuAC validation set by pre-training the BiDAF++ model on SQuAD 2.0 or CoQA. We tried combining the training data in all three datasets but achieved a validation F1 score of just 29.3 (compared to 50.2 after using just QuAC training data).

Dataset	Size	Rule	Hand	CNN
SQuAD	86.8k	30.5%	0.0%	69.5%
QuAC	65.2k	59.3%	1.5%	39.2%
CoQA	105.6k	57.1%	0.1%	42.8%
All	257.6k	48.7%	0.4%	50.9%

Table A1: Distribution of scheme adopted to classify questions in different datasets. "CNN" refers to the data-driven classifier. Roughly half the questions were classified using the rules described in Table 1.



Orson Welles

Breaking with the Federal Theatre Project in 1937, Welles and Houseman founded their own repertory company, which they called the Mercury Theatre. The name was inspired by the title of the iconoclastic magazine, The American Mercury. Welles was executive producer, and the original company included such actors as Joseph Cotten, George Coulouris, Geraldine Fitzgerald, Arlene Francis, Martin Gabel, John Hoyt, Norman Lloyd, Vincent Price, Stefan Schnabel and Hiram Sherman.

Q. What is Mercury Theatre?
A. Breaking with the Federal Theatre Project in 1937, Welles and Houseman founded their own repertory company, which they called the Mercury Theatre.

Q. What company did they form?
A. Federal Theatre Project

Q. When was the Federal Theatre Project founded?
A. 1937,

Q. Who started it?
A. Welles and Houseman founded their own repertory company, which they called the Mercury Theatre.

Q. Who was the producer?
A. Welles was executive producer, and the original company included such actors as Joseph Cotten, George Coulouris, Geraldine Fitzgerald, Arlene Francis, Martin Gabel,

Q. Why was it called the Federal Theatre?
A. The name was inspired by the title of the iconoclastic magazine, The American Mercury.

Q. What was the name of the iconoclastic magazine?
A. The American Mercury

Michael Phelps

Before the final of the 100-meter butterfly, US born Serbian swimmer Milorad Cavic caused a minor stir when he said it would be "good" if Phelps lost. "It'd be good for him if he loses. It would be nice if historians talk about Michael Phelps winning seven gold medals and losing the eighth to 'some guy.' I'd like to be that guy", Cavic said. Phelps responded, "When people say things like that, it fires me up more than anything." On August 16, Phelps won his seventh gold medal of the Games in the men's 100-meter butterfly, setting an Olympic record for the event with a time of 50.58 seconds and edging out his nearest competitor Cavic, by one hundredth (0.01) of a second.

Q. Why was he lost?
A. "It'd be good for him if he loses

Q. What did Phelps do on August 16?
A. On August 16, Phelps won his seventh gold medal of the Games in the men's 100-meter butterfly,

Q.Who did he win against?
A. 100-meter butterfly,

Q. Who is the Serbian swimmer?
A. US born Serbian swimmer Milorad Cavic

Q. Who did he lose?
A. Milorad Cavic

Q. When did he win a medal?
A. On August 16

Q. How many gold medals did he win?
A. Phelps won his seventh gold medal of the Games in the men's 100-meter butterfly,

Q. Who did he beat?
A. edging out his nearest competitor Cavic, by one hundredth (0.01) of a second.

Converge (band)

On February 25, 2003 Converge released their first official DVD, The Long Road Home. The DVD is modeled after band home videos such as Metallica's Cliff Em' All release. Deathwish Inc describes the DVD as a "two disc collection that is as energetic and exciting as the moments the release captures". The DVD also comes with a bonus disk that included three full live sets from the band.

Q. What did they do in 2003?
A. On February 25, 2003 Converge released their first official DVD, The Long Road Home.

Q. What was their first DVD?
A. On February 25, 2003 Converge released their first official DVD, The Long Road Home.

Q. When did they release this?
A. On February 25, 2003

Q. Where were the release?
A. The Long Road Home.

Q.What was the DVD about?
A. The DVD is modeled after band home videos such as Metallica's Cliff Em' All release.

Q. What other videos did they have?
A. Metallica's Cliff Em' All release.

Q. How many sets were from the band?
A. three full live sets from the bad

Q. What is Deathwish Inc?
A. Deathwish Inc describes the DVD as a "two disc collection that is as energetic and exciting as the moments the release captures".

Figure A1: Three SQUASH outputs generated by our system, showcasing the strengths and weaknesses described in Section 5.