Hierarchical Question Answering for Long Documents

Eunsol Choi
University of Washington
eunsol@cs.washington.edu

Daniel Hewlett, Alexandre Lacoste
Google Research
{dhewlett,allac}@google.com

Illia Polosukhin , Jakob Uszkoreit
Google Research
{ipolosukhin,usz}@google.com

Jonathan Berant
Tel Aviv University
joberant@cs.tau.ac.il

Abstract

Reading an article and answering questions about its content is a fundamental task for natural language understanding. While most successful neural approaches to this problem rely on recurrent neural networks (RNNs), training RNNs over long documents can be prohibitively slow. We present a novel framework for question answering that can efficiently scale to longer documents while maintaining or even improving performance. Our approach combines a coarse, inexpensive model for selecting one or more relevant sentences and a more expensive RNN that produces the answer from those sentences. A central challenge is the lack of intermediate supervision for the coarse model, which we address using reinforcement learning. Experiments demonstrate state-of-the-art performance on a challenging subset of the WIKIREADING dataset (Hewlett et al., 2016) and on a newly-gathered dataset, while reducing the number of sequential RNN steps by 88% against a standard sequence to sequence model.

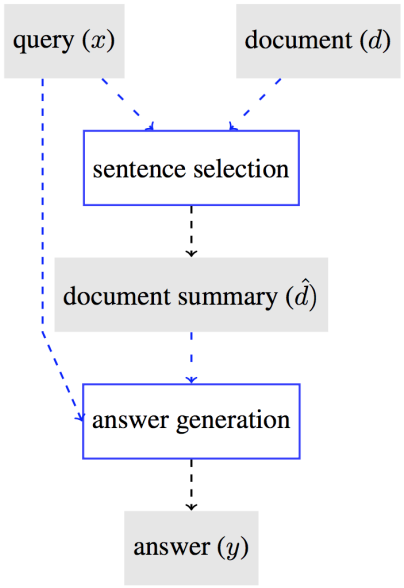


Figure 1: Hierarchical question answering: the model first selects relevant sentences that produce a document summary ($\hat{d}$) for the given query (x), and then generates an answer (y) based on the summary ($\hat{d}$).

1 Introduction

Reading a document and answering questions about its content are among the hallmarks of natural language understanding. Recently, interest in question answering (QA) from unstructured documents has rocketed along with the availability of large scale datasets for reading comprehension (Hermann et al., 2015; Hill et al., 2015; Rajpurkar et al., 2016; Onishi et al., 2016).

Current state-of-the-art approaches for QA over documents are based on Recurrent Neural Networks (RNNs) that encode the document and the question to determine the answer (Hermann et al., 2015; Chen et al., 2016; Kumar et al., 2016; Kadlec et al., 2016). While this allows the model to access all relevant information, it leads to slow models where the RNN runs sequentially over possibly thousands of tokens. In fact, most neural models for QA over documents truncate the documents and consider only a limited number of tokens (Miller et al., 2016; Hewlett et al., 2016). Inspired by studies (Masson, 1983) on how people answer questions by first skimming the docu-

ment, identifying the relevant parts, and then reading more carefully these parts to produce a final answer, we present in this paper a coarse-to-fine model for reading comprehension.

We propose a simple neural model that treats QA as a hierarchical process (see Figure 1). First, a fast model is used for sentence selection (Yu et al., 2014; Yang et al., 2016a), that is, to select a few sentences from the document that are relevant for answering the question. Then, a slower RNN is employed to produce the final answer from the selected sentences. Thus, the RNN is run over a fixed number of tokens, regardless of the length of the original document. Reducing the size of the input to the RNN results in a faster model compared to a model that treats the entire document as a single sequence.

We do not assume that the answer to a question appears verbatim in the input document (e.g., a question about the genre of a movie can often be answered even if not mentioned explicitly). Therefore, sentence selection is treated as a latent variable that is trained jointly with the answer generation model from the answer signal only. We explore both a hard attention sentence selection model, trained using REINFORCE (Williams, 1992), as well as a fully differentiable soft attention sentence selection model that is trained end-to-end. We find that in datasets where the answer frequently does not appear verbatim in the input document, training jointly the sentence selection and answer generation models improves performance compared to a pipeline approach in which sentence selection is trained separately through distant supervision. We also explore multiple sentence selection models, from a simple bag-of-words (BoW) model to a more expensive but still parallelizable convolutional model.

We evaluate our model on a subset of the recently published WIKIREADING dataset (Hewlett et al., 2016), focusing on examples where the input document is lengthy and sentence selection is challenging. We also evaluate our model on a new question answering dataset called WIKISUGGEST that contains more natural questions gathered from a search engine. Our contribution is a new modular framework and learning procedures for addressing question answering over long documents. The hierarchical framework captures document structure such as sentence boundaries and is able to deal with long documents or potentially multiple documents. Our framework is applicable to any model, be it neural or not. Experiments demonstrate that we improve performance compared to state-of-the-art models on the subset of WIKIREADING, and obtain comparable performance on WIKISUGGEST, while dramatically reducing the number of tokens processed by the RNN at training and test time.

d: s_1: The 2011 Joplin tornado was a catastrophic EF5-rated multiple-vortex tornado that struck Joplin, Missouri . . . . . .
s_4: It was the third tornado to strike Joplin since May 1971.
s_5: Overall, the tornado killed **158 people** (with an additional four indirect deaths), injured some 1,150 others, and caused damages amounting to a total of $2.8 billion.. . .

x: how many people died in joplin mo tornado
y: 158 people

Figure 2: A training example containing a document d, a question x and an answer y. In this example, the sentence s_5 is necessary for answering the question.

2 Problem Setting

Our task is defined as follows. Given a training set of question-document-answer triples $\{x^{(i)}, d^{(i)}, y^{(i)}\}_{i=1}^{N}$, our goal is to train a question answering model that will produce an answer y for a new question-document pair (x, d). A document d is a list of sentences $s_1, s_2, \ldots, s_{|d|}$, and we assume there is a latent subset of sentences from which the answer can be produced. Figure 2 illustrates a training example in which sentence s_5 is a key sentence for answering the question.

3 Data

We evaluate on WIKIREADING LONG, and a newly gathered WIKISUGGEST dataset.

WIKIREADING (Hewlett et al., 2016) is a recently released dataset that was automatically generated from Wikipedia and Wikidata, where the goal is as follows: given a Wikipedia page about an entity and a Wikidata property, such as PROFESSION, or GENDER, to infer the value based on the document. Due to the structure of Wikipedia, and the short length of most Wikipedia documents (median number of sentences: 9), the answer can usually be generated

	% answer string exists	avg # of answer match	% match first sent
WIKIREADING	47.1	1.22	35.4
WIKIREADING LONG	50.4	2.18	15.8
WIKISUGGEST	100	13.95	33.6

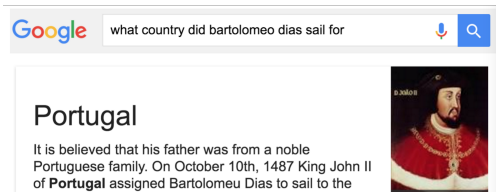
Table 1: Statistics on string matches of the answer y^* in the document for WIKIREADING, WIKIREADING LONG and WIKISUGGEST.

from the first few sentences. Thus, this dataset is not optimal for testing a sentence selection model compared to a model that just utilizes the first few sentences. Table 1 quantifies this intuition: We consider sentences containing the answer y^* as a proxy for sentences that are useful for answer generation, and report how often y^* appears in the document. Additionally, we report how frequently this proxy oracle sentence is the first sentence. We observe that in WIKIREADING, the answer appears in the document in 47.1% of the examples, and in 35.4% of the examples the match is on the first sentence, thus headroom for sentence selection models is limited.

To remedy that, we filter WIKIREADING and ensure a more even distribution of answers throughout the document. First, we prune documents with less than 10 sentences. Second, we only consider Wikidata properties for which Hewlett et al. (2016)'s best model obtains an accuracy that is lower than 60%. This step prunes out properties such as GENDER, GIVEN NAME and INSTANCE OF. This results in a dataset containing 1.97M examples, including properties such as GENRE, FILM EDITOR.(see Table 2 for further dataset statistics). On this subset, the answer appears in 50.4% of the examples, and it appears in the first sentence only 15.8% of the time (Table 1).

While WIKIREADING offers high-quality, large-scale data, the questions (Wikidata properties) are not phrased in natural language and are limited to 858 properties only. To model more realistic natural language queries, we collect the WIKISUGGEST dataset as follows (see Figure 3 for illustration).

We use the Google Suggest API to harvest natural language questions and submit them to Google Search. Whenever Google Search returns a box with a short answer whose source is Wikipedia (Figure 3), we create an example from the question, answer, and the Wikipedia document from which the



WIKISUGGEST Query	Answer
what year did virgina became a state	1788
general manager of smackdown	Theodore Long
minnesota viking colors	purple
coco martin latest movies	maybe this time
longest railway station in asia	Gorakhpur
son from modern family	Claire Dunphy
north dakota main religion	Christian
lands end' brand	Lands' End
wdsu radio station	WCBE

Figure 3: Example queries and answers of WIKISUGGEST.

answer was extracted. We evaluated the data quality by manually examining fifty examples. 54% of the examples were well formed question-answer pairs where we can ground answers in the given Wikipedia document, 20% contain answers without textual evidence in the document, and 26% contain incorrect question-answer pairs such as the last two examples in Figure 3.

Table 1 shows that the exact answer string is often missing from the document in WIKIREADING and WIKIREADING LONG. Extracted from manually curated Wikidata statements, WIKIREADING includes categorical properties such as GENDER and NATIONALITY. When the answer is not explicitly mentioned, it can usually be inferred from the document. On the other hand, in WIKISUGGEST, a missing answer string match often implies a spurious question-answer pair such as ('what time is half time in rugby', '80 minutes, 40 minutes'). Thus, we pruned question-answer pairs without exact answer string match for quality control. In both datasets, multiple sentences may contain the answer string. Most questions, however, are factoid and do not require multiple sentences to answer.

4 Model

Our model has two parts (see Figure 1). The first part is a fast sentence selection model (Section 4.1) that defines a probability distribution $p(s \mid x, d)$ over document sentences given the input question (x) and the document (d). The second part is a more costly

	# of unique query	# of examples	Avg. words / query	Avg. words / doc.	Avg. sents / doc.
WIKIREADING	858	16.03M	2.35	568.9	21.4
WIKIREADING LONG	239	1.97M	2.14	1200.7	49.9
WIKISUGGEST	3.47M	3.47M	5.03	5962.2	215.8

Table 2: Statistics for the WIKIREADING, WIKIREADING LONG and WIKISUGGEST datasets.

answer generation model (Section 4.3) that defines a probability distribution $p(y \mid x, \hat{d})$ over answers given a question and a document "summary" ($\hat{d}$) that focuses on the relevant parts of the document. The representation $\hat{d}$ is constructed with a soft attention model or a hard attention model (Section 4.2).

4.1 Sentence Selection Model

Following recent work on sentence selection for QA (Yu et al., 2014; Yang et al., 2016b), we build a feed-forward neural network model to define a probability distribution (attention) over the list of sentences $s_1, s_2, \ldots, s_{|d|}$, given the query x and document d. We consider three possible sentence representations for sentence selection: a bag of words (BoW) model, a chunking model, and a (parallelizable) convolutional model. These models are relatively efficient at dealing with larger quantities of text.

BoW Model A bag-of-words representation of an utterance amounts to averaging the word embeddings in that utterance. We explored two network structures for sentence scoring. The first is concatenation: the bag-of-words representation of the query is concatenated to the bag-of-words representation of the sentence s_l, which is then passed through a single layer feed-forward network.

$$\begin{aligned} h_l &= \text{BoW}(x)||\text{BoW}(s_l), \\ v_l &= v^\top \text{ReLU}(W h_l), \\ p(s = s_l \mid x, d) &= \text{softmax}(v_l), \end{aligned}$$

where $||$ is the concatenation operator, and the matrix W, the vector v and the word embeddings are learned parameters.

We also consider a dot product model without a non-linear interaction between the sentence and query, inspired by its strong performance in MT setting (Luong et al., 2015). The model passes the BoW representation of the sentence and query separately through a single layer feed-forward network and computes the dot product between the two output vectors.

$$\begin{aligned} h_x &= \text{ReLU}(W_q \text{ BoW}(x)), \\ h_l &= \text{ReLU}(W_d \text{ BoW}(s_l)), \\ p(s = s_l \mid x, d) &= \text{softmax}(h_x \cdot h_l), \end{aligned}$$

where the matrices W_q, W_d, the vectors v_q, v_d and the word embeddings are learned parameters.

Chunked BoW Model To get a more fine-grained granularity, one can split sentences into fixed size smaller chunks (seven tokens per chunk) and score each chunk separately (Miller et al., 2016). This could be beneficial if questions are answered by subsentential units that are lost in a BoW representation of an entire sentence. We split a sentence into a fixed number of chunks $(c_{l,1}, c_{l,2} \cdots, c_{l,J})$, generate a BoW representation for each chunk, and score it separately exactly as in the BoW model. We compute a probability distribution over chunks as in the BoW model, and compute sentence probabilities by marginalizing over all chunks belonging to a sentence. Let $p(c = c_{l,j} \mid x, d)$ be the probability distribution over all chunks from all sentences, then:

$$p(s = s_l \mid x, d) = \sum_{j=1}^{J} p(c = c_{l,j} \mid x, d),$$

with the same parameters as in the BoW model.

Convolutional Neural Network Model While our sentence selection model is designed to be fast, we explore a slightly more complex architecture by adding a convolutional neural network (CNN) that can compose the meaning of words. CNN is still more efficient than an RNN, since the convolutional filters can be computed in parallel. We follow previous work (Kim, 2014; Kalchbrenner et al., 2014) and add a single convolutional layer over the encoding $\text{enc}(x, s_l)$ with 100 filters of width 5. We use max-pooling to obtain a fixed-length representation for the sentence and query, and pass that representation through a single layer feed-forward network as in the BoW model to score each sentence.

4.2 Document summary

After computing an attention distribution over sentences with the sentence selection model, we create a summary that focuses on the document parts related to the question using a deterministic soft attention model or a stochastic hard attention model.

Hard Attention In the hard attention model, we sample a sentence $s_l \sim p(s \mid x, d)$ and fix the document summary $\hat{d}$ to be the chosen sentence. At test time, we choose the most probable sentence rather than sample.

To extend the document summary to contain further information, we also explore selecting K sentences from the document and define the summary to be the concatenation of the sampled sentences $\hat{d} = s_{l_1}||s_{l_2}||\ldots||s_{l_K}$.[1] Sampling prevents conventional back propagation and therefore we use the REINFORCE algorithm as discussed in Section 5.

Soft Attention The soft attention model (Bahdanau et al., 2015) generates the summary by computing a weighted average of sentences word by word according to $p(s \mid x, d)$. More explicitly, let $\hat{d}_m$ be the mth token of the document summary. Then, by fixing the length of every sentence to M tokens[2], the *blended* tokens are computed as follows:

$$\hat{d}_m = \sum_{l=1}^{|d|} p(s = s_l \mid x, d) \cdot s_{l,m},$$

where $m \in [1, \ldots, M]$, $s_{l,m}$ is the mth word in the lth sentence, and $|d|$ is the number of sentences in a document.

4.3 Answer Generation Model

State-of-the-art question answering models (Chen et al., 2016) use sequence-to-sequence models to encode the document and question and generate the answer. We focus on a developing a fast sentence selection model, and do not subscribe to a particular answer generation architecture. We choose to implement the state-of-the-art word-level sequence-to-sequence model with placeholders, described by Hewlett et al. (2016), and review it shortly here for completeness. This recurrent neural network model takes the query tokens, a separating token, and the document (or in out case, document summary) tokens as input and encodes them with a Gated Recurrent Unit (GRU) (Cho et al., 2014). Then, the answer is decoded with another GRU model with shared word embeddings parameters, which defines a distribution over answers $p(y \mid x, \hat{d})$.

5 Learning

We consider three approaches for learning the parameters of our model. The soft attention model is differentiable and is optimized using end-to-end learning. The hard attention model is non-differentiable and is optimized with the REINFORCE algorithm (Williams, 1992). Last, we consider a pipeline approach, where we use distant supervision to label sentences for selection and train a sentence selection model independently from an answer generation model. We use stochastic gradient descent with Adam (Kingma and Ba, 2015).

Distant Supervision While we do not have explicit supervision for the sentence selection model, we can define a simple heuristic for labeling sentences. We define the first sentence that has a full match of the answer string as the gold sentence, and if no such sentence exists, we define the first sentence to be the gold sentence.[3] By labeling gold sentences we can train the sentence selection and answer generation models independently. In this setup, y and s are given as targets, and s serves as the document summary. Hence, our objective function is:

$$\begin{aligned} J(\theta) &= \log p_\theta(y, s \mid x, d) \\ &= \log p_\theta(s \mid x, d) + \log p_\theta(y \mid s, x). \end{aligned}$$

At training time, we define the probability of the gold sentence $p(s = s_{\text{target}} \mid x, d) = 1$ and 0 for all other sentences. At test time we do not have access to the gold sentence and use

$$\arg\max_{s_l \in d} p(s = s_l \mid d, x)$$

directly for answer generation.

[1] To prevent re-sampling the same sentences, we mask the previously selected sentences and re-normalize $p_\theta(s \mid x, d)$ after sampling each example.

[2] Longer sentences are truncated and shorter ones are padded

[3] We experimented with learning the distribution over sentences that match the answer string, but this did not improve over matching the first sentence.

Reinforcement Learning In this setup, we consider that the target sentence is not provided and we use a reinforcement learning approach with two actions. The first action is sentence selection and the second is answer generation. Our goal is to select sentences that lead to a high reward. We define the reward for selecting a sentence to be the log probability of the correct answer given that sentence, that is, $R_\theta(s_l) = \log p_\theta(y = y^* \mid s_l, x)$. Then the learning objective is to maximize the expected reward:

$$J(\theta) = \sum_{s_l \in d} p_\theta(s = s_l \mid x, d) \cdot R_\theta(s_l)$$
$$= \sum_{s_l \in d} p_\theta(s = s_l \mid x, d) \cdot \log p_\theta(y = y^* \mid s_l, x)$$

Following REINFORCE (Williams, 1992), we approximate the gradient of the objective with a sample, $\hat{s} \sim p_\theta(s \mid x, d)$:

$$\nabla J(\theta) \approx \nabla \log p_\theta(y \mid \hat{s}, x) + \log p_\theta(y \mid \hat{s}, x) \cdot \nabla \log p_\theta(\hat{s} \mid x, d).$$

Extending the derivations above to sampling K sentences is straightforward and we omit it for brevity.

Training with REINFORCE is known to be unstable due to the high variance induced by the sampling. To reduce the variance, we use a curriculum learning strategy, gently transitioning from a distant supervision setting to a reinforcement learning setting, similar to the DAGGER algorithm (Ross et al., 2011). We define the probability of using the distant supervision objective at each global step as r^e, where r is the decay rate and e is the index of the current training epoch. We tuned $r \in [0.3, 1]$ for each sentence selection model and dataset combination on the development set.

Soft Attention We train the soft attention model by maximizing the conditioned log likelihood of the answer given the question and the document: $\log p_\theta(y^* \mid d, x)$. The model is fully differentiable and is trained end-to-end with back propagation.

6 Experiments

Experimental Setup We used 70% of the data for training, 10% for validation, and 20% for testing. We used the first 35 sentences in each document as input to the hierarchical models, where each sentence has a maximum length of 35 tokens. Similar to Miller et al (2016), we add the first five words in the document (typically the title of the article) at the end of each sentence sequence for WIKISUGGEST. We also add the sentence index as a one hot vector to the sentence representation. We fixed most hyperparameters for all models after tuning on the validation set.[4] The learning rate and gradient clipping coefficient were tuned separately on the validation set for each model with grid search over the values {0.0005, 0.001, 0.002, 0.004} and {0.5, 1.0}, respectively. We employed dropout (Srivastava et al., 2014) in the smaller WIKIREADING LONG dataset to avoid overfitting.

Dataset	Learning	Accuracy
WIKIREADING LONG	ORACLE	43.9
	PIPELINE	36.8
	REINFORCE (K=1)	40.1
	REINFORCE (K=2)	**42.2**
	SOFTATTEND	38.3
	FIRST	26.7
	FULL	40.1
WIKI SUGGEST	ORACLE	60.0
	PIPELINE	45.3
	REINFORCE (K=1)	45.4
	REINFORCE (K=2)	45.8
	SOFTATTEND	45.4
	FIRST	44.0
	FULL	**46.7**

Table 3: Answer prediction accuracy on the test set. K is the number of sentences in the document summary.

Evaluation Metrics Our main evaluation metric is answer accuracy, that is, the proportion of questions answered correctly. We do not perform any normalization on the answer string, and treat only exact string match to be correct.

For sentence selection, we report the accuracy of selecting the correct one. Since we do not know which sentence contains the answer, we report approximate accuracy by matching sentences that contain the answer string (y^*). For the soft attention model, we treat the sentence with the highest atten-

[4] Word embeddings dimension for RNN=256, vocabulary size=100,000 and a GRU state dimension=512. For the sentence selection model, we used concatenation in WIKIREADING LONG and dot product in WIKISUGGEST (see Section 4.1).

Dataset	Learning	Model	P@1	MRR
WIKI READING LONG	PIPELINE	CNN	35.1	41.9
		BOW	34.3	41.5
		CHUNKBOW	**37.0**	**42.9**
	REINFORCE	CNN	36.8	42.8
		BOW	35.8	42.3
		CHUNKBOW	36.9	42.8
	ORACLE		49.6	49.6
	FIRST		15.5	26.3
	SOFTATTEND (BOW)		35.2	41.3
WIKI SUGGEST	PIPELINE	CNN	48.7	57.5
		BOW	**52.8**	**61.3**
		CHUNKBOW	44.9	55.2
	REINFORCE	CNN	50.5	59.6
		BOW	52.6	61.1
		CHUNKBOW	46.6	57.0
	ORACLE		78.2	78.5
	FIRST		33.3	42.8
	SOFTATTEND (BOW)		39.0	49.3

Table 4: Approximate sentence selection accuracy for various sentence selection models and learning objectives on validation set.

tion probability as the predicted sentence. We report precision@1 (accuracy) and mean reciprocal rank (MRR) for this evaluation.

Models and Baselines The models PIPELINE, REINFORCE, and SOFTATTEND correspond to the three learning objectives in Section 5. We compare these models against the following baselines:

> **FULL** is an implementation of the best model by Hewlett et al. (2016). A word-level sequence-to-sequence model, which consumes the first 300 tokens of the document. We experimented with providing additional tokens to match the length of document available to hierarchical models, but the model failed to generalize to longer input sequences and showed degraded performance.
>
> **FIRST** always selects the first sentence of the document. The answer appears in the first sentence in 33% and 15% of documents in WIKISUGGEST and WIKIREADING LONG, respectively.
>
> **ORACLE** selects the first sentence with the answer string if it exists, or otherwise the first sentence in the document.

Finally, we also compare the accuracy of the sentence selection models (CNN, CHUNKBOW and BOW) described in Section 4.

Results Table 3 summarizes the answer accuracy for the two datasets. The proposed hierarchical models match or exceed the performance of FULL, while reducing the number of sequential RNN steps by almost an order of magnitude, from 300 to 35 (or 70 for K=2). When the answer appear after the first 300 tokens, FULL cannot access it unlike our models. While our approach incurs an additional cost for the sentence selection models, these models are fast and parallelizable, unlike the RNN model which takes most of the computation time. In WIKIREADING-LONG, our model based on REINFORCE outperforms all other models (excluding ORACLE, which has access to sentence labels at test time).

Jointly learning answer generation and sentence selection, REINFORCE outperforms PIPELINE, which relies on the noisy supervision signal for sentence selection, in WIKIREADING LONG. SOFT ATTEND also did better than PIPELINE, though by a lesser amount. In WIKISUGGEST, all learned models perform similarly, a bit lower than the FULL model. This is potentially explained by the approximate supervision for sentence selection, which exists for 78% of examples vs. 49% in WIKIREADING LONG. Results suggest that modeling the summary as a latent variable is more important when the answer string is absent in the document.

For hard attention, adding more information to the summary is achieved by sampling an additional sentence. Allowing REINFORCE to sample an additional sentence increased performance in both datasets, but by a larger amount in WIKIREADING LONG than WIKISUGGEST.[5] Additional sampling may allow recovery from mistakes in WIKIREADING LONG, where sentence selection performance was lower.

In both datasets, all models outperform the FIRST baseline, which heuristically selects the first sentence for all queries. Using the sentence with answer string match (ORACLE) improves the performance on both datasets, particularly on WIKISUGGEST.[6]

Comparing hard attention to soft attention, we observe that in general hard attention models were more effective, a pattern similar to results from caption generation (Xu et al., 2015). The attention dis-

[5] Sampling more did not help with pipeline methods.

[6] ORACLE may be able to fit the noise in the data, answering more spurious queries correctly.

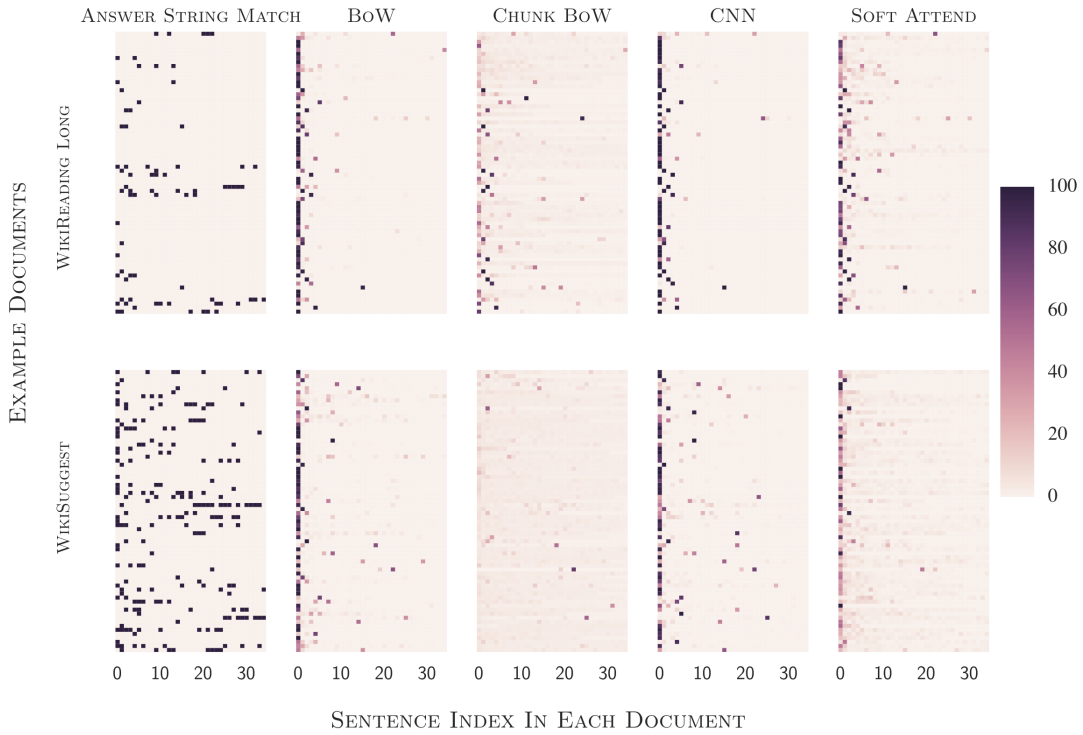


Figure 4: For a random subset of documents in the development set, we visualized the learned attention distribution over the sentences ($p(s_l|d, x)$).

tribution learned by the soft attention model is often less peaked (see Figure 4), which generates noisier summaries with irrelevant sentences. This likely explains the lower performance, since both datasets consist primarily of factoid questions that do not require reasoning over multiple sentences.

Table 4 reports sentence selection accuracy, according to our approximate evaluation. Precision@1 and MRR show the same trends. As in the answer accuracy, SoftAttend performed worse than Reinforce, while both outperform First. The simpler BoW performed the best in WikiSuggest, whereas the CNN and ChunkBoW model showed a slight improvement over BoW in WikiReading Long, where sentence selection is more challenging. While still representing an upper bound on performance, Oracle does not achieve a perfect score as some examples do not have an answer string match in the first 35 sentences.

Qualitative Analysis Figure 4 contains a visualization of the attention distribution over sentences, $p(s_l \mid d, x)$, for different learning procedures. The increased frequency of the answer string in WikiSuggest vs. WikiReading Long is evident in the leftmost plot. SoftAttend and ChunkBoW clearly distribute attention more evenly across the sentences compared to BoW and CNN.

	WikiReading Long	Wiki Suggest
No evidence in doc.	29	8
Error in answer generation	13	15
Noisy query & answer	0	24
Error in sentence selection	8	3

Table 5: Manual error analysis on 50 errors for Reinforce (K=1) for each dataset.

We categorized the primary reasons for the system's errors in Table 5 and present an example for each error type in Table 6. All examples are from Reinforce model with BoW sentence selection model. The most frequent source of error for WikiReading Long was the lack of evidence in the input document. While the dataset does not contain false answers, the document does not always provide supporting evidence (examples of properties without clues are elevation above sea level and sister). Interestingly, answer string match can still appear in the document as in the first example in Table 6: the answer 'Saint Petersburg' appears multiple times in the document (e.g., 4th

sentence). In both datasets, answer generation at times failed to generate the exact answer sentence even with the correct sentence selection. This was pronounced especially when the answer was long. In another example, the system generated 'Princess *UNK* of *UNK*'[7] instead of 'Princess Marie Louise of Bourbon-Parma'. For the automatically collected WIKISUGGEST dataset, noisy question-answer pairs were problematic, as discussed in Section 2. However, the models can frequently guess the answer correctly. Sentence selection was more challenging in WIKIREADING LONG, explaining why sampling two sentences improved performance more.

While the exact answer string appears in many documents, at times it was not mentioned and the model learned to generate the answer string sequence, as in the first correctly predicted example. The second example shows when our model successfully spots the relevant sentence without obvious clues. The third one shows a typical case, where the model correctly paid attention to the first sentence, while in the last example the model correctly spots the sentence far from the head of the document.

7 Related Work

In recent years, there has been an increasing interest in datasets that evaluate reading comprehension. The MCTest (Richardson et al., 2013) and BioProcess bank (Berant et al., 2014) are smaller-scale datasets focusing on common sense reasoning under a closed world assumption. bAbi (Weston et al., 2015) is a synthetic dataset with simulated questions to capture various aspects of reasoning. WikiQA is a well-curated, smaller-scale answer selection dataset. Jurczyk et al (2016) presented a crowdsourced dataset on Wikipedia which involves sentence selection and the SQuAD (Rajpurkar et al., 2016) dataset proposes answer selection, but covers shorter documents. Cloze-style question answering datasets (CNN (Hermann et al., 2015), Who did What (Onishi et al., 2016), and CBT (Hill et al., 2015)) are designed to assess machine comprehension but do not form valid questions.

Answer sentence selection, also referred as answer triggering, has been studied in the TREC QA dataset (Voorhees and Tice, 2000). Recently, neural network architectures (Wang and Nyberg, 2015; Severyn and Moschitti, 2015; dos Santos et al., 2016) achieved improvements over earlier feature-based models (Severyn and Moschitti, 2013). Recent work (Sultan et al., 2016) models answer sentence extraction and answer extraction jointly. Here we study answer sentence selection as a latent variable for a question answering task, and generate answer strings instead of selecting text spans.

Hierarchical attention models have been recently applied to tasks such as text categorization (Yang et al., 2016b), extractive summarization (Cheng and Lapata, 2016), machine translation (Ba et al., 2014), aspect sentiment classification (Lei et al., 2016), and vision (Lu et al., 2016; Xu et al., 2015). To the best of our knowledge, we are the first to directly use the document structure to improve question answering over documents.

Finally, our work is related to reinforcement learning literature and learning from noisy signals (Mintz et al., 2009; Hoffmann et al., 2011). The variants of hard and soft attention models were examined in the context of caption generation (Xu et al., 2015). Curriculum learning for question answering was investigated in Sachan and Xing (2016), but they focused on the ordering of training examples while we interpolate two supervision signals. Reinforcement learning gained popularity in natural language processing tasks such as coreference resolution (Clark and Manning, 2016), information extraction (Narasimhan et al., 2016), semantic parsing (Andreas et al., 2016) and textual games (Narasimhan et al., 2015; He et al., 2016).

8 Conclusion

We presented a hierarchical framework for QA over long documents that quickly focuses on the relevant portions of a document to answer the question. We demonstrated that our model is more efficient and can match or exceed state-of-the-art performance on two challenging reading comprehension datasets.

Our framework uses the document structure to handle long documents. In future work we would like to deepen the hierarchy by answering questions over multiple documents and utilize other structural clues such as paragraphs, titles and so on.

[7] *UNK* is the unknown token.

WIKIREADING LONG	Error Type (Query, Answer) System Output		No evidence in doc. (place_of_death, Saint Petersburg) Crimean Peninsula
	1 4 25	11.7 3.4 **63.6**	Alexandrovich Friedmann (also spelled Friedman or [Fridman] , Russian : . . . Friedmann was baptized into the Russian Orthodox Church as an infant , and lived much of his life in Saint Petersburg . . . . Friedmann died on September 16 , 1925 , at the age of 37 , from typhoid fever that he contracted while returning from a vacation in Crimean Peninsular .
	Error Type (Query, Answer) System Output		Error in sentence selection (position_played_on_team_speciality, power forward) point guard
	1 3	**37.8** 22.9	James Patrick Johnson (born February 20 , 1987) is an American professional basketball player for the Toronto Raptors of the National Basketball Association (NBA) .. . . Johnson was the starting power forward for the Demon Deacons of Wake Forest University
WIKISUGGEST	Error Type (Query, Answer) System Output		Error in answer generation (david blaine's mother, Patrice Maureen White) Maureen
	1 8 9	14.1 **22.6** 17.7	David Blaine (born David Blaine White; April 4, 1973) is an American magician, illusionist . . . The magic he offers operates on an uncommonly personal level . " Blaine was born and raised in, Brooklyn , New York the son of Patrice Maureen White (1946 – 1995) , His father was of half Puerto Rican and half Italian descent and his mother was of Russian Jewish ancestry.
	Error Type (Query, Answer) System Output		Noisy query & answer (what are dried red grapes called, dry red wines) Chardonnay
	1 2	2.8 **90.8**	Burgundy wine (French : Bourgogne or vin de Bourgogne) is wine made in the . . . The most famous wines produced here – those commonly referred to as "[Burgundies]" –are dry red wines made from Pinot noir grapes and white wines made from Chardonnay grapes .

Correctly Predicted Examples

WIKIREADING LONG	(Query, Answer)		(position_held, member of the National Assembly of South Africa)
	1 2	**98.4** 1.5	Anchen Margaretha Dreyer (born 27 March 1952) is a South African politician, a Member of Parliament for the opposition Democratic Alliance , and currently . . . Before being elected [unnopposed] to this position on 26 May 2014 she was first the Shadow . . .
	(Query, Answer)		(headquarters_locations, Solihull)
	1 4	13.8 **82.3**	LaSer UK is a provider of credit and loyalty programmes , operating in the UK and Republic . . . . . . The company 's operations are in Solihull and Belfast where it employs 800 people .
WIKISUGGEST	(Query, Answer)		(which state in nagpur, Maharashtra)
	1	**99.4**	Nagpur([Ngpur]) (pronunciation) is the second capital and the third largest city of the Indian state of Maharashtra after Mumbai and Pune . . . .
	(Query, Answer)		(avril lavigne husband, Chad Kroeger)
	1 20 23	17.6 13.4 **68.4**	Avril Ramona Lavigne ([vrłl] [lvin] / ; French pronunciation : ¡200b¿ ([avil] [lavi]) ;. . . In July 2006 , Lavigne married her boyfriend of two years , [Deryck] [Whibley] , lead singer and Lavigne married Nickelback frontman , Chad Kroeger , in 2013 . Avril Ramona Lavigne was . . .

Table 6: Example Outputs from REINFORCE (K=1) model with BOW sentence selection model. First column: sentence index (l). Second column: attention distribution $p_\theta(s_l|d, x)$. Last column: text s_l.

References

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. 2014. Multiple object recognition with visual attention. *The International Conference on Learning Representations*.

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

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the Conference of the Empirical Methods in Natural Language Processing*.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Association for Computational Linguistics*.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Proceedings of the Conference of the Empirical Methods in Natural Language Processing*.

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the Conference of the Empirical Methods in Natural Language Processing*.

Cícero Nogueira dos Santos, Ming Tan, Bing Xiang, and Bowen Zhou. 2016. Attentive pooling networks. *CoRR*, abs/1602.03609.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2016. Deep reinforcement learning with an unbounded action space. *Proceedings of the Conference of the Association for Computational Linguistics*.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*.

Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. 2016. Wikireading: A novel large-scale language understanding task over wikipedia. In *Proceedings of the Conference of the Association for Computational Linguistics*.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *The International Conference on Learning Representations*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics.

Tomasz Jurczyk, Michael Zhai, and Jinho D. Choi. 2016. SelQA: A New Benchmark for Selection-based Question Answering. In *Proceedings of the 28th International Conference on Tools with Artificial Intelligence*, ICTAI'16, San Jose, CA.

Rudolf Kadlec, Martin Schmid, Ondřej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 908–918, Berlin, Germany, August. Association for Computational Linguistics.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Proceedings of the Conference of the Empirical Methods in Natural Language Processing*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *The International Conference on Learning Representations*.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of the International Conference on Machine Learning*.

Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2016. Rationalizing neural predictions. *Proceedings of the Conference of the Empirical Methods in Natural Language Processing*.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. *arXiv preprint arXiv:1606.00061*.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September. Association for Computational Linguistics.

Michael EJ Masson. 1983. Conceptual processing of text during skimming and rapid sequential reading. *Memory & Cognition*, 11(3):262–274.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *Proceedings of the Conference of the Empirical Methods in Natural Language Processing*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011. Association for Computational Linguistics.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. 2015. Language understanding for text-based games using deep reinforcement learning. *Proceedings of the Conference of the Empirical Methods in Natural Language Processing*.

Karthik Narasimhan, Adam Yala, and Regina Barzilay. 2016. Improving information extraction by acquiring external evidence with reinforcement learning. *Proceedings of the Conference of the Empirical Methods in Natural Language Processing*.

Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2016. Who did what: A large-scale person-centered cloze dataset. *Proceedings of Empirical Methods in Natural Language Processing*.

P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the Conference of the Empirical Methods in Natural Language Processing*.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the Conference of the Empirical Methods in Natural Language Processing*.

Stéphane Ross, Geoffrey J Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics*.

Mrinmaya Sachan and Eric P Xing. 2016. Easy questions first? a case study on curriculum learning for question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Aliaksei Severyn and Alessandro Moschitti. 2013. Automatic feature engineering for answer selection and extraction. In *Proceedings of the Conference of the Empirical Methods in Natural Language Processing*.

Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 373–382. ACM.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Md. Arafat Sultan, Vittorio Castelli, and Radu Florian. 2016. A joint model for answer sentence ranking and answer extraction. *Transactions of the Association for Computational Linguistics*, 4:113–125.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207. ACM.

Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *Proceedings of the International Conference on Machine Learning*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2016a. Wikiqa: A challenge dataset for open-domain question answering. *Proceedings of the Conference of the Empirical Methods in Natural Language Processing*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016b. Hierarchical atten-

tion networks for document classification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep Learning for Answer Sentence Selection. In *NIPS Deep Learning Workshop*, December.