Coarse-to-Fine Question Answering for Long Documents

Eunsol Choi[†]
University of Washington
eunsol@cs.washington.edu

Daniel Hewlett, Jakob Uszkoreit
Google
{dhewlett,usz}@google.com

Illia Polosukhin[†]
XIX.ai
i@xix.ai

Alexandre Lacoste[†]
Element AI
allac@elementai.com

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

Abstract

We present a framework for question answering that can efficiently scale to longer documents while maintaining or even improving performance of state-of-the-art models. While most successful approaches for reading comprehension rely on recurrent neural networks (RNNs), running them over long documents is prohibitively slow because it is difficult to parallelize over sequences. Inspired by how people first skim the document, identify relevant parts, and carefully read these parts to produce an answer, we combine a coarse, fast model for selecting relevant sentences and a more expensive RNN for producing the answer from those sentences. We treat sentence selection as a latent variable trained jointly from the answer only using reinforcement learning. Experiments demonstrate the state of the art performance on a challenging subset of the WIKIREADING dataset (Hewlett et al., 2016) and on a new dataset, while speeding up the model by 3.5x-6.7x.

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 increased 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; Nguyen et al., 2016; Trischler et al., 2016a).

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; Xiong et al., 2016). While such models have access to all the relevant information, they are slow because the model needs to be run sequentially over possibly thousands of tokens, and the computation is not parallelizable. In fact, such models usually truncate the documents and consider only a limited number of tokens (Miller et al., 2016; Hewlett et al., 2016). Inspired by studies on how people answer questions by first skimming the document, identifying relevant parts, and carefully reading these parts to produce an answer (Masson, 1983), we propose a coarse-to-fine model for question answering.

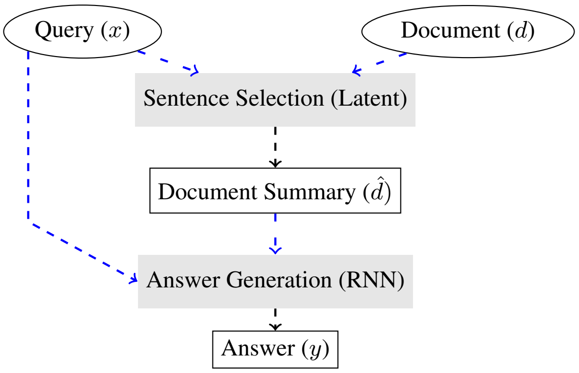


Figure 1: Hierarchical question answering: the model first selects relevant sentences that produce a document summary ($\hat{d}$) for the given query (x), and then generates an answer (y) based on the summary ($\hat{d}$) and the query x.

Our model takes a hierarchical approach (see Figure 1), where first a fast model is used to select a few sentences from the document that are relevant for answering the question (Yu et al., 2014; Yang et al., 2016a). Then, a slow RNN is employed to produce the final answer from the selected sentences. The RNN is run over a fixed number of tokens, regardless of the length of the document. Empirically, our model encodes the

[†]Work done while the authors were at Google.

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** . . ., injured some 1,150 others, and caused damages . . .

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 the WIKISUGGEST dataset. In this example, the sentence s_5 is necessary to answer the question.

text up to 6.7 times faster than the base model, which reads the first few paragraphs, while having access to four times more tokens.

A defining characteristic of our setup is that an answer does not necessarily appear verbatim in the input (the genre of a movie can be determined even if not mentioned explicitly). Furthermore, the answer often appears multiple times in the document in spurious contexts (the year '2012' can appear many times while only once in relation to the question). Thus, we treat sentence selection as a latent variable that is trained jointly with the answer generation model from the answer only using reinforcement learning. Treating sentence selection as a latent variable has been explored in classification (Yessenalina et al., 2010; Lei et al., 2016), however, to our knowledge, has not been applied for question answering.

We find that jointly training sentence selection and answer generation is especially helpful when locating the sentence containing the answer is hard. We evaluate our model on the WIKIREADING dataset (Hewlett et al., 2016), focusing on examples where the document is long and sentence selection is challenging, and on a new dataset called WIKISUGGEST that contains more natural questions gathered from a search engine.

To conclude, we present a modular framework and learning procedure for QA over long text. It captures a limited form of document structure such as sentence boundaries and deals with long documents or potentially multiple documents. Experiments show improved performance compared to the state of the art on the subset of WIKIREADING, comparable performance on other datasets, and a 3.5x-6.7x speed up in document encoding, while allowing access to much longer documents.

	% answer string exists	avg # of ans. match	% match first sent
WIKIREADING	47.1	1.22	75.1
WR-LONG	50.4	2.18	31.3
WIKISUGGEST	100	13.95	33.6

Table 1: Statistics on string matches of the answer y^* in the document. The third column only considers examples with answer match. Often the answer string is missing or appears many times while it is relevant to query only once.

2 Problem Setting

Given a training set of question-document-answer triples $\{x^{(i)}, d^{(i)}, y^{(i)}\}_{i=1}^{N}$, our goal is to learn a model that produces an answer y for a question-document pair (x, d). A document d is a list of sentences $s_1, s_2, \dots, s_{|d|}$, and we assume that the answer can be produced from a small latent subset of the sentences. Figure 2 illustrates a training example in which sentence s_5 is in this subset.

3 Data

We evaluate on WIKIREADING, WIKIREADING LONG, and a new dataset, WIKISUGGEST.

WIKIREADING (Hewlett et al., 2016) is a QA dataset automatically generated from Wikipedia and Wikidata: given a Wikipedia page about an entity and a Wikidata property, such as PROFESSION, or GENDER, the goal is to infer the target value based on the document. Unlike other recently released large-scale datasets (Rajpurkar et al., 2016; Trischler et al., 2016a), WIKIREADING does not annotate answer spans, making sentence selection more challenging.

Due to the structure and short length of most Wikipedia documents (median number of sentences: 9), the answer can usually be inferred from the first few sentences. Thus, the data is not ideal for testing a sentence selection model compared to a model that uses the first few sentences. Table 1 quantifies this intuition: We consider sentences containing the answer y^* as a proxy for sentences that should be selected, 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 verbatim in 47.1% of the examples, and in 75% of them the match is in the first sentence. Thus, the importance of modeling sentence selection is limited.

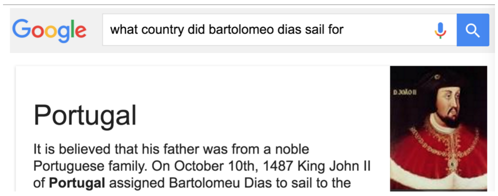
To remedy that, we filter WIKIREADING and ensure a more even distribution of answers throughout the document. We prune short docu-

	# of uniq. queries	# of examples	# of words / query	# of tokens / doc.
WIKIREADING	867	18.58M	2.35	489.2
WR-LONG	239	1.97M	2.14	1200.7
WIKISUGGEST	3.47M	3.47M	5.03	5962.2

Table 2: Data statistics.

ments with less than 10 sentences, and only consider Wikidata properties for which Hewlett et al. (2016)'s best model obtains an accuracy of less than 60%. This prunes out properties such as GENDER, GIVEN NAME, and INSTANCE OF.[1] The resulting WIKIREADING LONG dataset contains 1.97M examples, where the answer appears in 50.4% of the examples, and appears in the first sentence only 31% of the time. On average, the documents in WIKIREADING LONG contain 1.2k tokens, more tokens than those of SQuAD (average 122 tokens) or CNN (average 763 tokens) datasets (see Table 2). Table 1 shows that the exact answer string is often missing from the document in WIKIREADING. This is since Wikidata statements include properties such as NATIONALITY, which are not explicitly mentioned, but can still be inferred. A drawback of this dataset is that the queries, Wikidata properties, are not natural language questions and are limited to 858 properties.

To model more realistic language queries, we collect the WIKISUGGEST dataset as follows. 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 from Wikipedia (Figure 3), we create an example from the question, answer, and the Wikipedia document. If the answer string is missing from the document this 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 the exact answer string. We examined fifty examples after filtering and found that 54% were well-formed question-answer pairs where we can ground answers in the document, 20% contained answers without textual evidence in the document (the answer string exists in an irreleveant context), and 26% contain incorrect QA pairs such as the last two examples in Figure 3. The data collection was performed in May 2016.



what country did bartolomeo dias sail for

Portugal

It is believed that his father was from a noble Portuguese family. On October 10th, 1487 King John II of **Portugal** assigned Bartolomeu Dias to sail to 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.

4 Model

Our model has two parts (Figure 1): a fast sentence selection model (Section 4.1) that defines a distribution $p(s \mid x, d)$ over sentences given the input question (x) and the document (d), and a more costly answer generation model (Section 4.3) that generates an answer y given the question and a document summary, $\hat{d}$ (Section 4.2), that focuses on the relevant parts of the document.

4.1 Sentence Selection Model

Following recent work on sentence selection (Yu et al., 2014; Yang et al., 2016b), we build a feed-forward network to define a distribution over the sentences $s_1, s_2, \ldots, s_{|d|}$. We consider three simple sentence representations: a bag-of-words (BoW) model, a chunking model, and a (parallelizable) convolutional model. These models are efficient at dealing with long documents, but do not fully capture the sequential nature of text.

BoW Model Given a sentence s, we denote by $\text{BoW}(s)$ the bag-of-words representation that averages the embeddings of the tokens in s. To define a distribution over the document sentences, we employ a standard attention model (e.g., (Hermann et al., 2015)), where the BoW representation of the query is concatenated to the BoW representation of each sentence s_l, and then passed through a single layer feed-forward network:

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

[1] These three relations alone account for 33% of the data.

where $[;]$ indicates row-wise concatenation, and the matrix W, the vector v, and the word embeddings are learned parameters.

Chunked BoW Model To get more fine-grained granularity, we split sentences into fixed-size smaller chunks (seven tokens per chunk) and score each chunk separately (Miller et al., 2016). This is beneficial if questions are answered with sub-sentential units, by allowing to learn attention over different chunks. We split a sentence s_l into a fixed number of chunks $(c_{l,1}, c_{l,2} \ldots, c_{l,J})$, generate a BoW representation for each chunk, and score it exactly as in the BoW model. We obtain a distribution over chunks, and compute sentence probabilities by marginalizing over chunks from the same sentence. Let $p(c = c_{l,j} \mid x, d)$ be the distribution over 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 convolutional neural network (CNN) that can compose the meaning of nearby words. A CNN is still efficient, since all filters can be computed in parallel. Following previous work (Kim, 2014; Kalchbrenner et al., 2014), we concatenate the embeddings of tokens in the query x and the sentence s_l, and run a convolutional layer with F filters and width w over the concatenated embeddings. This results in F features for every span of length w, and we employ max-over-time-pooling (Collobert et al., 2011) to get a final representation $h_l \in \mathbb{R}^F$. We then compute $p(s = s_l \mid x, d)$ by passing h_l through a single layer feed-forward network as in the BoW model.

4.2 Document Summary

After computing attention over sentences, we create a summary that focuses on the document parts related to the question using deterministic soft attention or stochastic hard attention. Hard attention is more flexible, as it can focus on multiple sentences, while soft attention is easier to optimize and retains information from multiple sentences.

Hard Attention We sample a sentence $\hat{s} \sim p(s \mid x, d)$ and fix the document summary $\hat{d} = \hat{s}$ to be that sentence during training. At test time, we choose the most probable sentence. To extend the document summary to contain more information, we can sample without replacement K sentences from the document and define the summary to be the concatenation of the sampled sentences $\hat{d} = [\hat{s}_1; \hat{s}_2; \ldots; \hat{s}_K]$.

Soft Attention In the soft attention model (Bahdanau et al., 2015) we compute a weighted average of the tokens in the sentences 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 $s_{l,m}$ is the mth word in the lth sentence ($m \in \{1, \ldots, M\}$).

As the answer generation models (Section 4.3) take a sequence of vectors as input, we average the tokens at the word level. This gives the hard attention an advantage since it samples a "real" sentence without mixing words from different sentences. Conversely, soft attention is trained more easily, and has the capacity to learn a low-entropy distribution that is similar to hard attention.

4.3 Answer Generation Model

State-of-the-art question answering models use RNN models to encode the document and question and selects the answer. We focus on a hierarchical model with fast sentence selection, and do not subscribe to a particular answer generation architecture.

Here we implemented the state-of-the-art word-level sequence-to-sequence model with placeholders, described by Hewlett et al. (2016). This models can produce answers that does not appear in the sentence verbatim. This model takes the query tokens, and the document (or 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, defining a distribution over answers $p(y \mid x, \hat{d})$. In this work, we modified the original RNN: the word embeddings for the RNN decoder input, output and original word embeddings are shared.

[2] Long sentences are truncated and short ones are padded.

5 Learning

We consider three approaches for learning the model parameters (denoted by θ): (1) We present a pipeline model, where we use distant supervision to train a sentence selection model independently from an answer generation model. (2) The hard attention model is optimized with REINFORCE (Williams, 1992) algorithm. (3) The soft attention model is fully differentiable and is optimized end-to-end with backpropagation.

Distant Supervision While we do not have an explicit supervision for sentence selection, we can define a simple heuristic for labeling sentences. We define the gold sentence to be the first sentence that has a full match of the answer string, or the first sentence in the document if no full match exists. By labeling gold sentences, we can train sentence selection and answer generation independently with standard supervised learning, maximizing the log-likelihood of the gold sentence and answer, given the document and query. Let y^* and s^* be the target answer and sentence , where s^* also serves as the document summary. The objective is to maximize:

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

Since at test time we do not have access to the target sentence s^* needed for answer generation, we replace it by the model prediction $\arg\max_{s_l \in d} p_\theta(s = s_l \mid d, x)$.

Reinforcement Learning Because the target sentence is missing, we use reinforcement learning where our action is sentence selection, and our goal is to select sentences that lead to a high reward. We define the reward for selecting a sentence as 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:

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

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

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

Sampling K sentences is similar and omitted for brevity.

Training with REINFORCE is known to be unstable due to the high variance induced by sampling. To reduce variance, we use curriculum learning, start training with distant supervision and gently transition to reinforcement learning, similar to DAGGER (Ross et al., 2011). Given an example, we define the probability of using the distant supervision objective at each step as r^e, where r is the decay rate and e is the index of the current training epoch.[3]

Soft Attention We train the soft attention model by maximizing the log likelihood of the correct answer y^* given the input question and document $\log p_\theta(y^* \mid d, x)$. Recall that the answer generation model takes as input the query x and document summary $\hat{d}$, and since $\hat{d}$ is an average of sentences weighted by sentence selection, the objective is differentiable and is trained end-to-end.

6 Experiments

Experimental Setup We used 70% of the data for training, 10% for development, and 20% for testing in all datasets. 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) at the end of each sentence sequence for WIKISUGGEST. We add the sentence index as a one hot vector to the sentence representation.

We coarsely tuned and fixed most hyperparameters for all models. The word embedding dimension is set to 256 for both sentence selection and answer generation models. We used the decay rate of 0.8 for curriculum learning. Hidden dimension is fixed at 128, batch size at 128, GRU state cell at 512, and vocabulary size at 100K. For CNN sentence selection model, we used 100 filters and set filter width as five. The initial learning rate and gradient clipping coefficients for each model are tuned on the development set. The ranges for learning rates were 0.00025, 0.0005, 0.001, 0.002, 0.004 and 0.5, 1.0 for gradient clipping coefficient.

[3] We tuned $r \in [0.3, 1]$ on the development set.

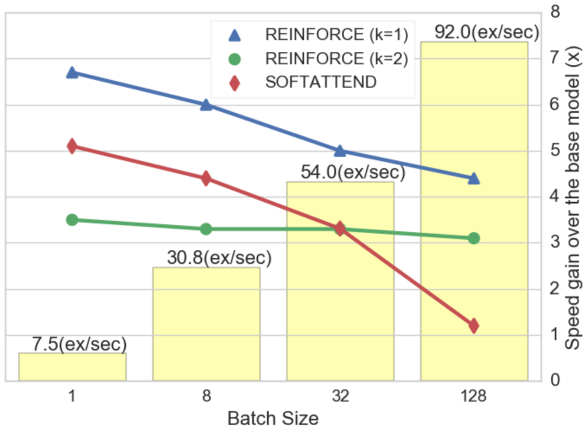


Figure 4: Runtime for document encoding on an Intel Xeon CPU E5-1650 @3.20GHz on WIKIREADING at test time. The boxplot represents the throughput of BASE and each line plot shows the proposed models' speed gain over BASE. Exact numbers are reported in the supplementary material.

We halved the learning rate every 25k steps. We use the Adam (Kingma and Ba, 2015) optimizer and TensorFlow framework (Abadi et al., 2015).

Evaluation Metrics Our main evaluation metric is answer accuracy, the proportion of questions answered correctly. For sentence selection, 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 probability as the predicted sentence.

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

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.

BASE is the re-implementation of the best model by Hewlett et al. (2016), consuming the first 300 tokens. We experimented with providing additional tokens to match the length of document available to hierarchical models, but this performed poorly.[4]

ORACLE selects the first sentence with the answer string if it exists, or otherwise the first sentence in the document.

Dataset	Learning	Accuracy
WIKIREADING LONG	FIRST	26.7
	BASE	40.1
	ORACLE	43.9
	PIPELINE	36.8
	SOFTATTEND	38.3
	REINFORCE (K=1)	40.1
	REINFORCE (K=2)	**42.2**
WIKI SUGGEST	FIRST	44.0
	BASE	**46.7**
	ORACLE	60.0
	PIPELINE	45.3
	SOFTATTEND	45.4
	REINFORCE (K=1)	45.4
	REINFORCE (K=2)	45.8
WIKIREADING	FIRST	71.0
	HEWLETT ET AL. (2016)	71.8
	BASE	**75.6**
	ORACLE	74.6
	SOFTATTEND	71.6
	PIPELINE	72.4
	REINFORCE (K=1)	73.0
	REINFORCE (K=2)	73.9

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

Answer Accuracy Results Table 3 summarizes answer accuracy on all datasets. We use BOW encoder for sentence selection as it is the fastest. The proposed hierarchical models match or exceed the performance of BASE, while reducing the number of RNN steps significantly, from 300 to 35 (or 70 for K=2), and allowing access to later parts of the document. Figure 4 reports the speed gain of our system. While throughput at training time can be improved by increasing the batch size, at test time real-life QA systems use batch size 1, where REINFORCE obtains a 3.5x-6.7x speedup (for K=2 or K=1). In all settings, REINFORCE was at least three times faster than the BASE model.

All models outperform the FIRST baseline, and utilizing the proxy oracle sentence (ORACLE) improves performance on WIKISUGGEST and WIKIREADNG LONG. In WIKIREADING, where the proxy oracle sentence is often missing and documents are short, BASE outperforms ORACLE.

Jointly learning answer generation and sentence selection, REINFORCE outperforms PIPELINE, which relies on a noisy supervision signal for sentence selection. The improvement is larger in WIKIREADING LONG, where the approximate supervision for sentence selection is missing for 51% of examples compared to 22% of examples in WIKISUGGEST.[5]

On WIKIREADING LONG, REINFORCE outper-

[4] Our numbers on WIKIREADING outperform previously reported numbers due to modifications in implementation and better optimization.

[5] The number is lower than in Table 1 because we cropped sentences and documents, as mentioned above.

Dataset	Learning	Model	Accuracy
WikiReading Long	Pipeline	CNN	70.7
		BoW	69.2
		ChunkBoW	**74.6**
	Reinforce	CNN	74.2
		BoW	72.2
		ChunkBoW	**74.4**
	First		31.3
	SoftAttend (BoW)		70.1
WikiSuggest	Pipeline	CNN	62.3
		BoW	**67.5**
		ChunkBoW	57.4
	Reinforce	CNN	64.6
		BoW	**67.3**
		ChunkBoW	59.3
	First		42.6
	SoftAttend (BoW)		49.9

Table 4: Approximate sentence selection accuracy on the development set for all models. We use Oracle to find a proxy gold sentence and report the proportion of times each model selects the proxy sentence.

forms all other models (excluding Oracle, which has access to gold labels at test time). In other datasets, Base performs slightly better than the proposed models, at the cost of speed. In these datasets, the answers are concentrated in the first few sentences. Base is advantageous in categorical questions (such as Gender), gathering bits of evidence from the whole document, at the cost of speed. Encouragingly, our system almost reaches the performance of Oracle in WikiReading, showing strong results in a limited token setting.

Sampling an additional sentence into the document summary increased performance in all datasets, illustrating the flexibility of hard attention compared to soft attention. Additional sampling allows recovery from mistakes in WikiReading Long, where sentence selection is challenging.[6] Comparing hard attention to soft attention, we observe that Reinforce performed better than SoftAttend. The attention distribution learned by the soft attention model was often less peaked, generating noisier summaries.

Sentence Selection Results Table 4 reports sentence selection accuracy by showing the proportion of times models selects the proxy gold sentence when it is found by Oracle. In WikiReading Long, Reinforce finds the approximate gold sentence in 74.4% of the examples where the the answer is in the document. In WikiSuggest performance is at 67.5%, mostly due to noise in the data. Pipeline performs slightly better as it is directly trained towards our noisy evaluation. However, not all sentences that contain the answer are useful to answer the question (first example in Table 6). Reinforce learned to choose sentences that are likely to generate a correct answer rather than proxy gold sentences, improving the final answer accuracy. On WikiReading Long, complex models (CNN and ChunkBoW) outperform the simple BoW, while on WikiSuggest BoW performed best.

	WR 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 from the development set for Reinforce (K=1).

Qualitative Analysis We categorized the primary reasons for the errors in Table 5 and present an example for each error type in Table 6. All examples are from Reinforce with BoW sentence selection. The most frequent source of error for WikiReading Long was lack of evidence in the 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, the answer string can still appear in the document as in the first example in Table 6: 'Saint Petersburg' appears in the document (4th sentence). Answer generation at times failed to generate the answer even when the correct sentence was selected. This was pronounced especially in long answers. For the automatically collected WikiSuggest dataset, noisy question-answer pairs were problematic, as discussed in Section 3. However, the models frequently guessed the spurious answer. We attribute higher proxy performance in sentence selection for WikiSuggest to noise. In manual analysis, sentence selection was harder in WikiReading Long, explaining why sampling two sentences improved performance.

In the first correct prediction (Table 6), the model generates the answer, even when it is not in the document. The second example shows when our model spots the relevant sentence without obvious clues. In the last example the model spots a sentence far from the head of the document.

Figure 5 contains a visualization of the atten-

[6] Sampling more help pipeline methods less.

WikiReading Long (WR 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 . . . 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 Peninsula .
	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 (WS)	Error Type (Query, Answer) System Output		Error in answer generation (david blaine's mother, Patrice Maureen White) Maureen
	1 8	14.1 **22.6**	David Blaine (born David Blaine White; April 4, 1973) is an American magician, illusionist . . . Blaine was born and raised in, Brooklyn , New York the son of Patrice Maureen White . . .
	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 . . . are dry red wines made from Pinot noir grapes . . .

Correctly Predicted Examples

WR Long	(Query, Answer)		(position_held, member of the National Assembly of South Africa)
	1	**98.4**	Anchen Margaretha Dreyer (born 27 March 1952) is a South African politician, a Member of Parliament for the opposition Democratic Alliance , and currently . . .
	(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 .
WS	(Query, Answer)		(avril lavigne husband, Chad Kroeger)
	1 23	17.6 **68.4**	Avril Ramona Lavigne ([vrłl] [lvin] / ; French pronunciation : ¡200b¿ ([avil] [lavi]) ;. . . Lavigne married Nickelback frontman , Chad Kroeger , in 2013 . Avril Ramona Lavigne was . . .

Table 6: Example outputs from REINFORCE (K=1) 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.

tion 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.

7 Related Work

There has been substantial interest in datasets for reading comprehension. MCTest (Richardson et al., 2013) is a smaller-scale datasets focusing on common sense reasoning; bAbi (Weston et al., 2015) is a synthetic dataset that captures various aspects of reasoning; and SQuAD (Rajpurkar et al., 2016; Wang et al., 2016; Xiong et al., 2016) and NewsQA (Trischler et al., 2016a) are QA datasets where the answer is a span in the document. Compared to Wikireading, some datasets covers shorter passages (average 122 words for SQuAD). Cloze-style question answering datasets (Hermann et al., 2015; Onishi et al., 2016; Hill et al., 2015) assess machine comprehension but do not form questions. The recently released MS MARCO dataset (Nguyen et al., 2016) consists of query logs, web documents and crowd-sourced answers.

Answer sentence selection is studied with the TREC QA (Voorhees and Tice, 2000), WikiQA (Yang et al., 2016b) and SelQA (Jurczyk et al., 2016) datasets. Recently, neural networks models (Wang and Nyberg, 2015; Severyn and

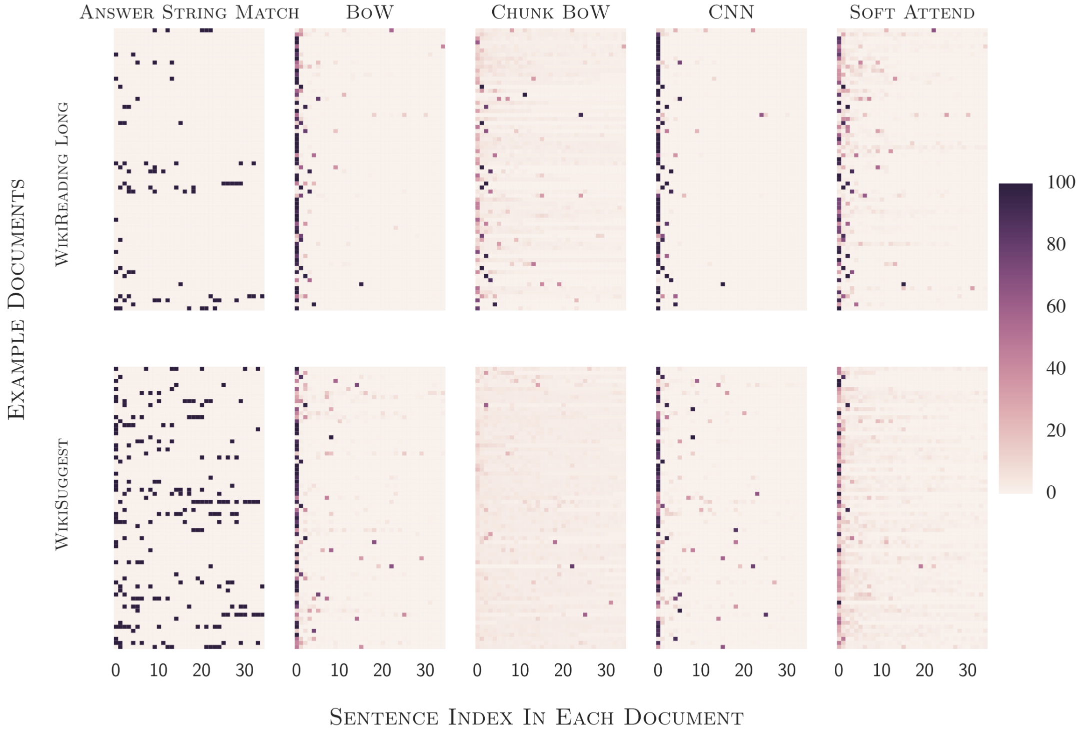


Figure 5: For a random subset of documents in the development set, we visualized the learned attention over the sentences ($p(s_l|d,x)$).

Moschitti, 2015; dos Santos et al., 2016) achieved improvements on TREC datsaet. Sultan et al. (2016) optimized the answer sentence extraction and the answer extraction jointly, but with gold labels for both parts. Trischler et al. (2016b) proposed a model that shares the intuition of observing inputs at multiple granularities (sentence, word), but deals with multiple choice questions. Our model considers answer sentence selection as latent and generates answer strings instead of selecting text spans, and we found that WIKIREADING dataset suits our purposes best with some pruning, which still provided 1.97 million examples compared to 2K questions for TREC dataset.

Hierarchical models which treats sentence selection as a latent variable have been applied text categorization (Yang et al., 2016b), extractive summarization (Cheng and Lapata, 2016), machine translation (Ba et al., 2014) and sentiment analysis (Yessenalina et al., 2010; Lei et al., 2016). To the best of our knowledge, we are the first to use the hierarchical nature of a document for QA.

Finally, our work is related to the reinforcement learning literature. Hard and soft attention were examined in the context of caption generation (Xu et al., 2015). Curriculum learning was investigated in Sachan and Xing (2016), but they focused on the ordering of training examples while we combine supervision signals. Reinforcement learning recently gained popularity in 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 coarse-to-fine framework for QA over long documents that quickly focuses on the relevant portions of a document. In future work we would like to deepen the use of structural clues and answer questions over multiple documents, using paragraph structure, titles, sections and more. Incorporating coreference resolution would be another important direction for future work. We argue that this is necessary for developing systems that can efficiently answer the information needs of users over large quantities of text.

Acknowledgement

We appreciate feedbacks from Google colleagues. We also thank Yejin Choi, Kenton Lee, Mike Lewis, Mark Yatskar and Luke Zettlemoyer for comments on the earlier draft of the paper. The last author is partially supported by Israel Science Foundation, grant 942/16.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. http://tensorflow.org/.

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* .

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*.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research (JMLR)* 12:2493–2537.

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*. http://arxiv.org/abs/1506.03340.

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* .

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*. San Jose, CA, ICTAI'16. https://arxiv.org/abs/1606.08513.

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)*. Association for Computational Linguistics, Berlin, Germany, pages 908–918. http://www.aclweb.org/anthology/P16-1086.

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* .

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* .

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* .

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Workshop in Advances in Neural Information Processing Systems*.

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. 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*. ACM, pages 373–382.

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.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016a. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830* .

Adam Trischler, Zheng Ye, Xingdi Yuan, Jing He, Phillip Bachman, and Kaheer Suleman. 2016b. A parallel-hierarchical model for machine comprehension on sparse data. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* .

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*. ACM, pages 200–207.

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*.

Zhiguo Wang, Haitao Mi, Wael Hamza, and Radu Florian. 2016. Multi-perspective context matching for machine comprehension. *arXiv preprint arXiv:1612.04211* .

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.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604* .

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

Ainur Yessenalina, Yisong Yue, and Claire Cardie. 2010. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1046–1056.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep Learning for Answer Sentence Selection. In *NIPS Deep Learning Workshop*. http://arxiv.org/abs/1412.1632.