CHIME: Cross-passage Hierarchical Memory Network for Generative Review Question Answering

Junru Lu[1], **Gabriele Pergola**[1], **Lin Gui**[1], **Binyang Li**[2] **and Yulan He**[1]
[1]Department of Computer Science, University of Warwick, UK
[2]School of Information Science and Technology,
University of International Relations, Beijing, China
{Junru.Lu, Gabriele.Pergola, Lin.Gui, Yulan.He}@warwick.ac.uk
byli@uir.edu.cn

Abstract

We introduce CHIME, a cross-passage hierarchical memory network for question answering (QA) via text generation. It extends XLNet (Yang et al., 2019) introducing an auxiliary memory module consisting of two components: the *context memory* collecting cross-passage evidence, and the *answer memory* working as a buffer continually refining the generated answers. Empirically, we show the efficacy of the proposed architecture in the multi-passage generative QA, outperforming the state-of-the-art baselines with better syntactically well-formed answers and increased precision in addressing the questions of the AmazonQA review dataset. An additional qualitative analysis revealed the interpretability introduced by the memory module.

1 Introduction

With the development of large-scale pre-trained Language Models (LMs) such as BERT (Devlin et al., 2018), XLNet (Yang et al., 2019), and T5 (Raffel et al., 2019), tremendous progress has been made in Question Answering (QA). Fine tuning pre-trained LMs on task-specific data has surpassed human performance on QA datasets such as SQuAD (Rajpurkar et al., 2016) and NewsQA (Trischler et al., 2016). Nevertheless, most existing QA systems largely deal with factoid questions and assume a simplified setup such as multiple-choice questions, retrieving spans of text from given documents, and filling in the blanks. However, in many more realistic situations such as online communities, people tend to ask '*descriptive*' questions (e.g., '*How to improve the sound quality of echo dot?*'). Answering such questions requires the identification, linking, and integration of relevant information scattered over long-form multiple documents for the generation of free-form answers.

We are particularly interested in developing a QA system for questions from e-shopping communities using customer reviews. Compared to factoid QA systems, building a review QA system faces the following challenges: (1) as opposed to extractive QA where answers can be directly extracted from documents or multiple-choice QA where systems only need to make a selection over a set of pre-defined answers, review QA needs to gather evidence across multiple documents and generate answers in free-form text; (2) while factoid QA mostly centres on 'entities' and only needs to deal with limited types of questions, review QA systems are often presented with a wide variety of '*descriptive*' questions; (3) customer reviews may contain contradictory opinions. Review QA systems need to automatically identify the most prominent opinion given a question for answer generation.

In our work here, we focus on the AmazonQA dataset (Gupta et al., 2019), which contains a total of 923k questions and most of the questions are associated with 10 reviews and one or more answers. We propose a novel Cross-passage Hierarchical Memory Network named CHIME to address the aforementioned challenges. Regular neural QA models search answers by interactively comparing the question and supporting text, which is in line with human cognition in solving factoid questions (Zheng et al., 2019; Guthrie and Mosenthal, 1987). While for opinion questions, the cognition process is deeper: reading larger scale and more complex texts, building cross-text comprehension, continually refine the opinions, and finally form the answers (Zheng et al., 2019). Therefore, CHIME is designed to maintain

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/.

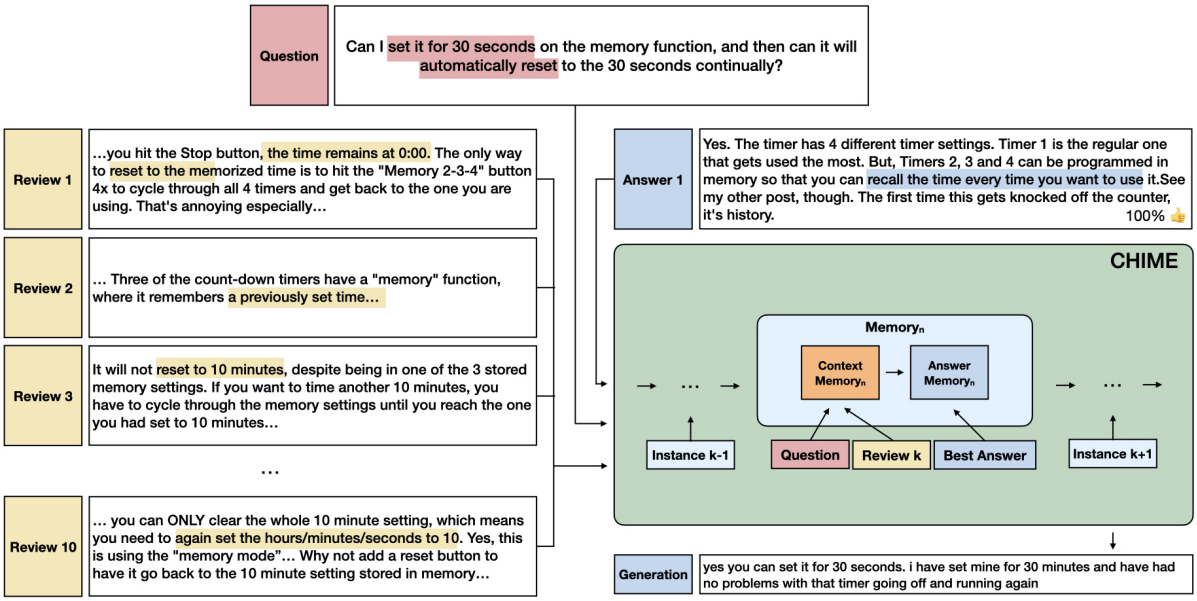

Figure 1: Illustration of the review QA task and the general idea of CHIME. The example question (the top box) is paired with 10 reviews (left panel) and one or more answers (right upper panel). CHIME is trained on the (Question, Review, Answer) triplet. During testing, CHIME is presented with a question and 10 related reviews and generates an answer (right bottom box). Both reviews and answers in this example contain contradictory information as highlighted by colors, while the question contains complex sub-questions. CHIME is able to identify relevant evidence and generate clear answers.

hierarchical dual memories to closely simulates this cognition process. In this model, a *context memory* dynamically collect cross-passage evidences, an *answer memory* stores and continually refines answers generated as CHIME reads supporting text in a sequential manner. Figure 1 illustrates the setup of our task and an example output generated from CHIME. The top box shows a question extracted from our test set while the left panel and the right upper panel show the related 10 reviews and the paired 4 actual answers. We can observe that the question can be decomposed into complex sub-questions and both reviews and answers contain contradictory information. However, CHIME can deal with such information effectively and generate appropriate answers as shown in the right-bottom box.

In summary, we have made the following contributions:

- We propose a novel Cross-passage HIerarchical MEmory Network (CHIME) for review QA. Compared with many multi-passage QA models, CHIME does not rely on explicit helpful ranking information of supporting reviews, but can capture cross-passage contextual information and effectively identify the most prominent opinion in reviews.
- CHIME reads reviews sequentially, overcoming the input length limitation affecting most of the existing transformer-based systems, and brings some interpretability for these "black box" models.
- Experimental results on the AmazonQA dataset show that CHIME outperforms a number of competitive baselines in terms of the quality of answers generated.

2 Related Work

Our work is related to the following three lines of research:

Opinion/Review Question-Answering In Opinion or Review QA, questions may concern about finding subjective personal experiences or opinions of certain products and services. The Amazon QA dataset was first released in (McAuley and Yang, 2016) which contains 1.4 million questions (and answers) and 13 million reviews on 191 thousand products collected from Amazon product pages. They developed a Mixture of Expert (MoE) model which automatically detects whether a review of a product is relevant to a given query. In their subsequent work, Wan and McAuley (2016) noticed that users tend to ask for

subjective information and answers might also be highly subjective and possibly contradictory. They, therefore, built a new dataset with 800 thousand questions and over 3 million answers from Amazon, in which each question is paired with multiple answers, and extended their previous MoE model with subjective information such as review rating scores and reviewer's bias incorporated. But they found that subjective information is only effective in predicting 'yes/no' answers to binary questions and does not help in distinguishing 'true' answers from alternatives in open-ended '*descriptive*' questions. More recently, Yu and Lam (2018) only focused on the yes/no questions in the Amazon QA dataset (McAuley and Yang, 2016) and trained a binary answer prediction model by leveraging latent aspect-specific representations of both questions and reviews learned by an autoencoder. Gao et al. (2019) focused on factual QA in e-commerce and proposed a Product-Aware Answer Generator that combines reviews and product attributes for answer generation, and uses a discriminator to determine whether the generated answer contains question-related facts. Xu et al. (2019a) proposed an extractive review-based QA task and manually created just over 2,500 questions and annotated the corresponding answer spans in less than 1,000 reviews relating to laptops and restaurants from the review data of SemEval 2016 Task 5[1]. They first jointly fine-tuned BERT for answer span detection, aspect extraction and aspect sentiment classification on the SemEval 2016 Task 5 data, and then post-trained BERT on over 3 million unlabelled Amazon and Yelp reviews in order to fuse domain knowledge, and also on SQuAD 1.1 (Rajpurkar et al., 2016) in order to gain task-relevant but out-of-domain knowledge. Gupta et al. (2019) created a subset from the Amazon QA product review dataset (McAuley and Yang, 2016), consisting of 923k questions with 3.6M answers and 14M reviews on 156k Amazon products. They trained an answerability classifier from 3,297 question-context pairs labeled by Mechanical Turk and used it to classify answerability for the whole dataset. They then converted the dataset into a span-based format by heuristically creating an answer span from reviews that best answers a question based on users' actual answers, and trained R-Net (Wang et al., 2017), which uses a gated self-attention mechanism and pointer networks, to predict answer boundaries. There are few studies using generative models to deal with opinion/review-based QA.

Multi-passage QA There are mainly two types of methods for multi-passage QA. One is to use retrieval-based methods to first identify text passages that are most likely to contain answer information, and then perform QA on the extracted text passages which are essentially considered as a single passage. The other one is to separately run single-passage QA over each passage, obtaining multiple answer candidates, and then determine the best answer through mutual verification among the answers.

Examples in the first type of methods include S-NET (Tan et al., 2018), Multi-passage BERT (Wang et al., 2019), and Masque (Nishida et al., 2019). These models require supporting text passages to be explicitly annotated. S-NET (Tan et al., 2018) follows an extraction-then-synthesis framework. First, relevant passages are extracted from context using a variant of R-NET (Wang et al., 2017), which learns to rank passages and extract the most possible evidence span from the selective passage; then, the evidence-notated selective passage is used for the GRU decoder synthesizing answers. In Multi-passage BERT (Wang et al., 2019), two independent BERTs were used to perform multi-passage QA. One BERT takes the question and a text passage as input and then uses the hidden states of the CLS token to train a classifier to determine if the text passage is relevant to the given question. The other BERT is used for extracting candidate answers from relevant text passages. The Masque model (Nishida et al., 2019) is a generative reading comprehension approach based on multi-source abstractive summarization. Masque uses a joint-learning framework, comprising of a question answerability classifier, a passage ranker, and an answer generator. At each step of answer generation, the decoder chooses a word from the mixture of three distributions derived from a vocabulary, from the question and associated multiple passages. A representative example of the second type of methods is V-Net (Wang et al., 2018). The main assumption of V-Net is that correct answers often appear in multiple documents with high frequency and similarity, and wrong answers are usually different from each other. Therefore, V-Net builds a mutual verification mechanism between all answer candidates, which are separately extracted from different passages, to select the best final answer.

Most existing approaches require explicit annotations of supporting text passages in order to train

[1] http://alt.qcri.org/semeval2016/task5/

multi-passage QA models in a supervised way. In our setup here, supporting review passages to a question was unsupervised ranked by BM25, which may introduce noises to QA model training and poses a more significant challenge.

Memory Network Memory network has been first proposed to model the relation between a story and a query for QA systems (Weston et al., 2014; Sukhbaatar et al., 2015). Apart from its application in QA, memory networks have also achieved great successes in other NLP tasks, such as machine translation (Maruf and Haffari, 2017), sentiment analysis (Fan et al., 2018), visual question answering (Xiong et al., 2016), social networks (Fu et al., 2020), and summarization (Kim et al., 2018). The main idea of memory networks is to use the attention mechanism to assign different weights to text passages so as to identify the most relevant passages for answer generation (Weston et al., 2014). Kumar et al. (2016) proposed a gated memory network to represent facts in different iterations during the learning process to verify the potentially related passages to generate an answer. Gui et al. (2017) used a convolutional architecture to capture attention signals in memory networks. Xu et al. (2019b) leveraged the memory network as an information retrieval system to search possible entities in knowledge bases for complex questions. Chen et al. (2019) used the memory network to verify items in knowledge bases as passages and then generate answers. Generally speaking, existing memory-network-based QA methods mainly focus on using memory networks to weigh and derive representations of question-aware text passages and knowledge entities for answer generation. We instead explore a novel structure of a hierarchical memory network composing of both *context* and *answer* memories for better capturing review context and generating more appropriate answers.

3 Cross-passage Hierarchical Memory Network (CHIME)

In this section, we first define the review QA task and then present our proposed Cross-passage Hierarchical Memory Network (CHIME).

3.1 Task Formulation

We focus on generative QA with multiple reviews and develop our model based on the AmazonQA dataset (Gupta et al., 2019) in which most of the questions is paired with multiple answers and the top 10 most relevant text snippets as supporting passages extracted from the associated reviews by BM25 (Robertson and Zaragoza, 2009). In addition, each question is annotated if it is answerable based on the top 10 review snippets, and each answer is accompanied with response votes. The review QA task can be defined as: given an answerable question $\boldsymbol{x}^q = \{x_1^q, x_2^q, \cdots, x_{N_q}^q\}$, K supporting reviews with k-th review represented as $\boldsymbol{x}^{r_k} = \{x_1^{r_k}, x_2^{r_k}, \cdots, x_{N_r}^{r_k}\}$, a model is asked to generate an answer $\hat{\boldsymbol{y}} = \{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_{N_a}\}$, where N_q, N_r and N_a denote the length of a question, a review and an answer, respectively. In training phase, L answers with l-th answer represented as $\boldsymbol{y}^{\boldsymbol{a_l}} = \{y_1^{a_l}, y_2^{a_l}, \cdots, y_{N_a}^{a_l}\}$ and corresponding response votes $\boldsymbol{v}^{\boldsymbol{a_l}} = \{v_+^{a_l}, v^{a_l}\}$ are provided, where $v_+^{a_l}$ denotes the number of positive votes and v^{a_l} denotes the number of all votes, and $0 \leq v_+^{a_l} \leq v^{a_l}$.

3.2 CHIME

In this paper, we propose a Cross-passage HIerarchical MEmory Network (CHIME) for review question answering. As has been shown in (Petroni et al., 2019), pre-trained LMs can be used as implicit knowledge bases, making them suitable for language generation. Hence, in this paper, we leverage the XLNet (Yang et al., 2019), which combines advantages of autoregressive and autoencoder models. Based on our task formulation, CHIME is designed to maximize the probability $p(\hat{\boldsymbol{y}}|\boldsymbol{x}^q, \boldsymbol{x}^{r_1} \cdots \boldsymbol{x}^{r_K})$ of generating an answer given a question and its associated K reviews in multi-passage review QA. The overall architecture of CHIME is shown in Figure 2. Given a question paired with K text passages, we create K training instances with each one consisting of the question, a text passage, and the best answer chosen by the helpfulness votes assigned by users. Each training instance is fed into an XLNet encoder to derive hidden representations, which will be used to update two memories. In particular, the *context memory* is updated when seeing more text passages and the *answer memory* is continuously refined with the answer generated from each (*question, text passage*) pair. CHIME has the following characteristics: (1) the use of

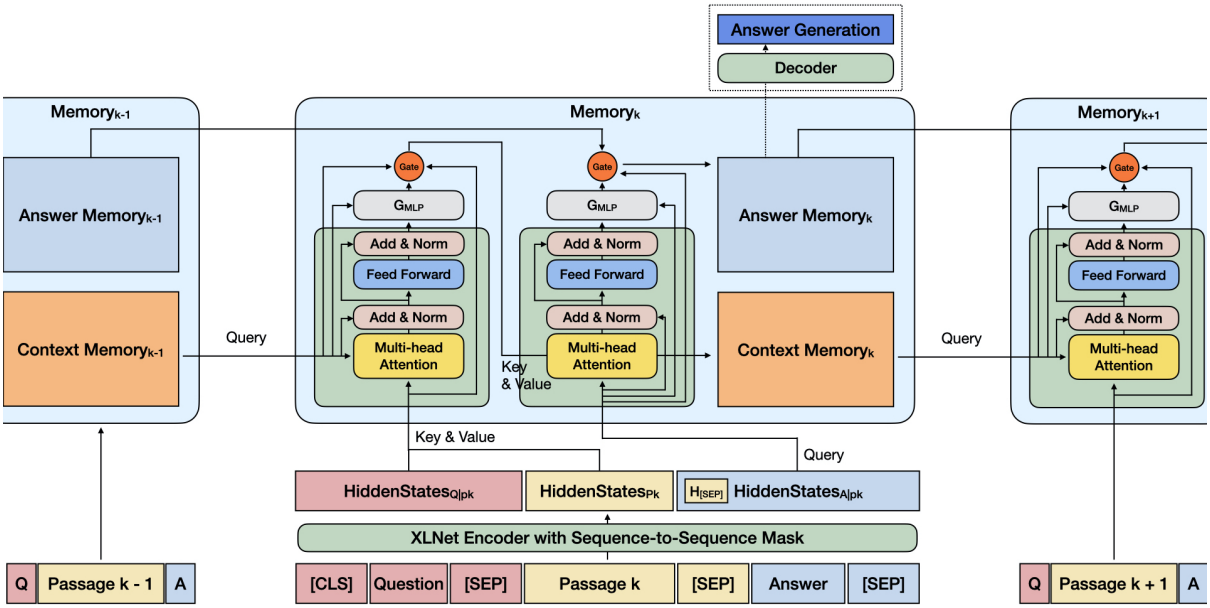

Figure 2: The architecture of Cross-passage Hierarchical Memory Network (CHIME). The model reads multiple reviews in a sequential manner. When reading the instance k consists of the question, k-th review, and the gold answer, the model first derive hidden states of the instance k from the XLNet encoder, then use the hidden states of context part update the *context memory* (the left part of the Memory$_k$). With the newly updated context memory, CHIME then be able to use the hidden states of the answer part to update the *answer memory* (the middle part of the Memory$_k$). After reading the last review, the *answer memory* will be input to the decoder and get a final answer generation (the top dotted frame).

a pre-trained XLNet as an encoder instead of traditional recurrent neural networks as the pre-trained LMs captures rich background knowledge and is more suitable for encoding semantic meanings of questions and review documents; (2) the proposal of the cross-passage *context memory* mechanism to perform the reading of review passages in a sequential manner to deal with multiple text passages more effectively, which avoids the massive memory costs required to read all supporting passages in one go; (3) the use of the *answer memory* to gradually refine the generated answer for a question after reading more text passages. Figure 2 shows the general architecture of CHIME, which consists of three key components: the XLNet encoder for encoding a question, a review, and an answer, the cross-passage hierarchical memory mechanism, and the decoder for answer generation.

XLNet Encoder The XLNet Encoder in CHIME is a vanilla XLNet encoder with special Seq2Seq masks introduced in UniLM (Dong et al., 2019), which is essentially a concatenation of a standard pre-trained LM encoder and a pre-trained LM decoder. With the Seq2Seq masks, we are able to train an encoder for an encoder-decoder task. In specific, for each question paired with K text passages, we create K training instances with each one consisting of the triple `(question, passage, answer)`. We add the special token `[CLS]` at the beginning and insert `[SEP]` as a separator between every two elements in the triple and add another `[SEP]` at the end. In addition, we treat `([CLS] Question [SEP] Passage [SEP])` as Part 1 and `(Answer [SEP])` as Part 2. The Seq2Seq masks are designed in a way such that all tokens in Part 1 attend to each other, and tokens in Part 2 attend to any tokens in Part 1, but only preceding tokens in Part 2. Let $\boldsymbol{y}^{a_g}$ be the gold-standard answer selected for current training instance, $\boldsymbol{x}_*^{r_k}$ be the whole input sequence of instance k; N_x be the length of $\boldsymbol{x}_*^{r_k}$, which keeps the same across all text passages; d be the dimension of hidden size, and $H^{r_k} \in \mathbb{R}^{N_x \times d}$ be the contextual hidden states of the encoder:

$$\begin{aligned} \boldsymbol{x}_*^{r_k} &= \big[\text{[CLS]} \quad \boldsymbol{x}^q \quad \text{[SEP]} \quad \boldsymbol{x}^{r_k} \quad \text{[SEP]} \quad \boldsymbol{y}^{a_g} \quad \text{[SEP]}\big] \\ H^{r_k} &= \text{XLNetEncoder}\big(E_t(x_*^{r_k}) + E_s(x_*^{r_k}) + E_p(x_*^{r_k})\big) \end{aligned} \tag{1}$$

where $E_t(\cdot)$, $E_s(\cdot)$ and $E_p(\cdot)$ denote token embeddings, segment embeddings and position embeddings respectively. Here we use an interval segment embedding $[E_t^A, E_t^B, E_t^A]$ to distinguish question, passage and answer other than the usual two-segment embedding in regular XLNet. As answers are only available during the training phase, training XLNet for the encoder-decoder task can be considered as fine-tuning pre-trained XLNet on our corpus in order to learn a better XLNet encoder.

Cross-passage hierarchical memory mechanism Hidden states of Part 1 and Part 2 are used to initialize and update *context memory* and *answer memory* respectively. Here the last `[SEP]` token in Part 1 is removed and added as the start token of Part 2 from this stage onwards for language generation purpose. Memory update is accomplished by taking a weighted aggregation of the previously retained memory and the current hidden state using a forget gate. The gate is obtained by using an MLP layer with a memory-specific Transformer encoder (Vaswani et al., 2017), which is composed of a multi-head scaled dot product attention sublayer and a position-wise fully connected feed forward network sublayer. When receiving the hidden states derived from XLNet encoder, CHIME first use the states of Part 1 to update *context memory*, then hierarchically use the newly updated *context memory* with the states of Part 2 to update *answer memory*. Let N_{S_1} and N_{S_2} be the length of Part 1 and Part 2, respectively, which are kept the same across different text passages; $H_c^{r_k} \in \mathbb{R}^{N_{S_1} \times d}$ be the hidden states of the context part, which refers to the question and a text passage; $H_a^{r_k} \in \mathbb{R}^{N_{S_2} \times d}$ be the hidden states of the answer part; $M_c^{r_k} \in \mathbb{R}^{N_{S_1} \times d}$ and $M_a^{r_k} \in \mathbb{R}^{N_{S_2} \times d}$ be the updated *context memory* and *answer memory* respectively after reading k-th passage:

$$
\begin{aligned}
Z_c^{r_k} &= \text{TransformerEncoder}(M_c^{r_{k-1}}, H_c^{r_k}) & Z_a^{r_k} &= \text{TransformerEncoder}(H_a^{r_k}, M_c^{r_k}) \\
G_c^{r_k} &= \sigma(W_{mc}^{r_k} M_c^{r_{k-1}} + W_{zc}^{r_k} Z_c^{r_k} + b_c^{r_k}) & G_a^{r_k} &= \sigma(W_{ha}^{r_k} H_a^{r_k} + W_{za}^{r_k} Z_a^{r_k} + b_a^{r_k}) \\
M_c^{r_k} &= G_c^{r_k} M_c^{r_{k-1}} + (1 - G_c^{r_k}) H_c^{r_k} & M_a^{r_k} &= G_a^{r_k} H_a^{r_k} + (1 - G_a^{r_k}) M_a^{r_{k-1}}
\end{aligned}
$$

where $Z_c^{r_k} \in \mathbb{R}^{N_{S_1} \times d}$ and $Z_a^{r_k} \in \mathbb{R}^{N_{S_2} \times d}$ denote the normalized attention output from the Transformer encoder, $G_c^{r_k} \in \mathbb{R}_{[0,1]}^{N_{S_1} \times d}$ and $G_a^{r_k} \in \mathbb{R}_{[0,1]}^{N_{S_2} \times d}$ denote the forget gate. $W_{mc}^{r_k} \in \mathbb{R}^{N_{S_1} \times d}$, $W_{zc}^{r_k} \in \mathbb{R}^{N_{S_1} \times d}$, $b_c^{r_k} \in \mathbb{R}^{N_{S_1}}$, $W_{ha}^{r_k} \in \mathbb{R}^{N_{S_2} \times d}$, $W_{za}^{r_k} \in \mathbb{R}^{N_{S_2} \times d}$ and $b_a^{r_k} \in \mathbb{R}^{N_{S_2}}$ are all trainable parameters. The two memories are initialized by taking the hidden states after reading the first review text passage of a question: $M_c^{r_1} = H_c^{r_1}, M_a^{r_1} = H_a^{r_1}$.

Decoder and Loss Function The answer probability $p(\hat{\boldsymbol{y}})$ over all V tokens of the whole vocabulary is generated by adding a softmax layer on the top of the answer memory:

$$
p(\hat{\boldsymbol{y}}) = \text{Softmax}(W_{ma} M_a^{r_K} + b_a) \tag{2}
$$

where $W_{ma} \in \mathbb{R}^{d \times V}$ and $b_a \in \mathbb{R}^V$ are trainable. The training loss of each sample is the cross entropy loss of the predicted answer $\hat{\boldsymbol{y}}$ and gold-standard answer $\boldsymbol{y}$:

$$
L = -\frac{1}{N_a} \sum_{n=1}^{N_a} y_n \log \hat{y_n} \tag{3}
$$

4 Experiments

In this section, we first introduce the dataset used in our experiments, the baselines for comparison, and the evaluation metrics employed, followed by a discussion over the obtained results and a few examples generated using the different approaches presented.

4.1 Settings

Dataset We built our dataset[2] from AmazonQA (Gupta et al., 2019). We only focused on more difficult '*descriptive*' questions and filter out non-answerable or 'yes/no' questions. We kept questions with 10 review snippets, sorting in descending order of relevance to the question. In the original dataset, 96% of

[2]Our dataset and codes are available at: `https://github.com/LuJunru/CHIME`.

the answerable '*descriptive*' questions are paired with 10 reviews. For each question, we only selected the best answer with the highest positive response rate. We further removed URL links from question, review, and answer text. The filtered dataset contains 365k samples in the training set, 47k samples in the validation set and 48k samples in the testing set. We set the maximum tokenized lengths of questions, reviews, and answers to 40, 124, and 82, respectively, which cover 95% of our samples.

Parameters setup The hidden size of BERT-base and XLNet-base is 768. The corresponding vocabulary sizes are 28,996 and 32,000. For CHIME, the inner Transformer encoders are 1-block vanilla Transformer, which contains an 8-heads multi-head attention and a feed-forward network with 2048 inner state size. The optimizer of all neural baselines is AdamW (Loshchilov and Hutter, 2018) with $\beta 1 = 0.9, \beta 2 = 0.999$, and $\epsilon = 1e-06$. Except for parameters of bias and layer normalization, all other training parameters are decayed with a rate of 0.95. The gradients of all parameters are clipped to the maximum norm 1.0. The learning rate is increased linearly from 0 to 1e-5 in the first 20% total training steps and then linearly decreased to 0.

Baselines We developed two heuristic baselines as well as three neural baselines:

- **Random Sentence**. Given a question, select a random sentence from paired reviews as an answer.
- **Sentence Retrieval**. First, convert each question and each sentence of its paired reviews into sentence embeddings using BERT, then retrieve the sentences with the highest cosine similarity with the question as the selective answer. The sentence length of both heuristic baselines is 120.
- **BERT+summary**. Directly using BERT (Devlin et al., 2018) for generative QA is difficult since it is memory demanding to deal with multiple reviews in one go. We instead first generate an extractive summary of reviews using Textrank (Mihalcea and Tarau, 2004), then feed a question and its associated review summary into BERT for answer generation.
- **XLNet+summary**. Although XLnet is theoretically capable of dealing with the text of unlimited length as it adopts the segmentation mechanism from Transformer XL (Dai et al., 2019), and could potentially process at once the concatenation of all the passages paired with a question, the computational requirements easily became rather prohibitive, and in practice is often not feasible to simultaneously deal with multiple long reviews with limited computational resources. Therefore, we take a similar summary-then-QA approach for XLNet.
- **XLNet+V-Net**. We follow the mutual verification mechanism proposed in V-Net (Wang et al., 2018) for answer post-processing. In particular, after XLNet generates candidate answers given individual reviews, mutual verification is conducted by calculating the average attention value of the current candidate answer with all the other answers. The one with the highest value is the final answer.

Due to the limitations of our computing resources, we have to use regular versions of large-scale pre-trained LMs and a subset of the original data. We use the BERT-base and the XLNet-base from Huggingface[3]. Both the neural baselines and our proposed CHIME are trained with 25% randomly selected data from our constructed dataset, which consists of 92k samples, comparable to popular large-scale datasets such as MS Marco (100k) (Nguyen et al., 2016) and HotpotQA (113k) (Yang et al., 2018). For all neural models, we train for 3 epochs and use the beam search with size 3 over the best models to generate answers from decoder probability distributions. In testing phase, 1k samples are extracted randomly for answer generation and evaluation.

Metrics We use ROUGE-L (Lin, 2004) and BLEU (Papineni et al., 2002) to evaluate the lexical similarity between the gold-standard and the model generated answers. To measure the semantic similarity, we use BertScore[4] (Zhang et al., 2019), which first computes the pairwise cosine similarity among all the tokens in the candidate and reference answers, and then greedily match them to get the highest similarity score for the sentence pair. BLEURT[5] (Sellam et al., 2020) is a text generation quality evaluation framework that uses BLEU, ROUGE and BertScore and other indicators as multi-task joint training through fine-tuning BERT. We use BLEURT as a comprehensive metric to evaluate both the lexical and semantic

[3] https://github.com/huggingface/transformers/blob/master/src/transformers

[4] https://github.com/Tiiiger/bert_score

[5] https://github.com/google-research/bleurt

similarities. A higher BLEURT score means that the generated sentence is both lexically and semantically closer to the ground truth. As each question is paired with multiple ground-truth answers, for BertScore and BleuRT, we finally consider the pair obtaining the maximum score.

4.2 Results

Model	Bleu-1	Bleu-2	Rouge-L F1	BertScore	BleuRT
Heuristic Baselines					
Random Sentence	25.189	8.996	15.669	0.103	-1.043
Retrieval Sentence	24.895	8.848	15.393	0.099	-1.040
Neural Baselines					
BERT + Summary	31.404	14.494	16.856	-0.027	-1.376
XLNet + Summary	32.037	14.018	20.484	0.162	**-0.866**
XLNet + V-Net	31.950	14.465	20.807	0.176	-0.952
CHIME	**33.103**	**14.947**	**21.512**	**0.185**	-0.949
CHIME-c	29.552	14.202	20.831	0.174	-0.982
CHIME-a	31.361	14.142	20.988	0.177	-0.976

Table 1: Evaluation results of CHIMEs and baselines. The answers generated by CHIMEs are superior in terms of lexical and semantics evaluations. CHIME-c removes the *context memory* and makes use of just the *answer memory*, in which the *answer memory* is updated not by *context memory* but by current context hidden states. In contrast, CHIME-a removes the *answer memory* and makes use of just the *context memory*, in which we remove the MLP sublayer for *answer memory* and directly feed the output of middle transformer encoder to the final decoder as shown in Figure 2.

Table 1 reports the evaluations of 1k selective samples from the testing set. The answers generated by CHIME exhibit an overall improved quality reflected by lexical and semantic evaluations outperforming all baselines. This validates the efficacy of combining the *context* and the *answer memory* to generate coherent answers when processing multiple passages, containing possibly contradictory opinions. CHIME-c is an ablated version of CHIME that only uses the *answer memory*, which is updated without the link from the *context memory* $M_c^{r_k}$ but using the current context hidden states $H_c^{r_k}$. The comparison of CHIME-c with CHIME demonstrates the importance of the cross-passage evidence collection. Similarly, CHIME-a is another ablated version that makes use of the only *context memory*, in which we link $Z_a^{r_k}$ from the *answer memory*'s encoder for the final decoding. The performance gap between CHIME-a and CHIME corroborate the relevance of a gradual answer refinement.

4.3 Qualitative analysis

As a case study, we analyze the example reported in Figure 1. We first compare the quality of the answers generated by different models and then illustrate a breakdown of the CHIME's generative process when iteratively reading different reviews. The gradual generative process provides some explicit interpretability of cross-passage evidence collection and sequential answer refinement.

In Table 2 we compare a few answers generated using different models[6]. Answers returned by either randomly selecting a sentence from review text passages or by retrieving a sentence from passages which is most similar to a given question are clearly not directly addressing the question. The poor quality of the answer returned by the BERT+Summary model, off-topic and ill-grammatical structure, shows the limitation of simply using the out-of-the-box BERT in text generation. The XLNets are able to generate some reliable answers, which is much better than the BERT+Summary. However, the two XLNet models mistakenly uses "30 mins" to replace the key term "30 seconds", which weakens the credibility of the answer. Compared with XLNets, the CHIME generates syntactically well-formed answer with better coherency and fluency.

[6]More example outputs are presented in Appendix.

Question:	Can I set it for 30 seconds on the memory function, and then can it will automatically reset to the 30 seconds continually?
Answer 1:	Yes, you can reset the times you set and reuse them.
Answer 2:	No...thats why it sucks...
Answer 3:	Do not know
Answer 4:	yes. the timer has 4 different timer settings. timer 1 is the regular one that gets used the most. but, timers 2, 3 and 4 can be programmed in memory so that you can recall the time every time you want to use it. see my other post, though. the first time this gets knocked off the counter, it's history.
Random Sentence:	the only way to reset to the memorized time is to hit the "memory 2-3-4" button 4x to cycle through all 4 timers and get back to the one you are using.
Retrieval Sentence:	but, and here's the major flaw, once the timer counts down to 0:00, beeps, and you hit the stop button, the time remains at 0:00.
BERT + Summary:	i have not have a little good. i don't have a little light. i'm not sure, but you can.
XLNet + Summary:	i have no problem with this. it works for 30 mins, but the timer will reset to 0, but not to a point where you can adjust your timer.
XLNet + V-Net:	yes, it will reset to 30 seconds, but i don't know if it would work for 30 minutes or not? i have no idea about this issue but.
CHIME:	yes you can set it for 30 seconds. i have set mine for 30 minutes and have had no problems with that timer going off and running again.

Table 2: Comparison of example answers. CHIME outperforms all the five baselines in generating more reliable answer. XLNets provide readable but noisy, incomplete and ill-understandable answers. All other baselines including the BERT one are not able to generate readable answers.

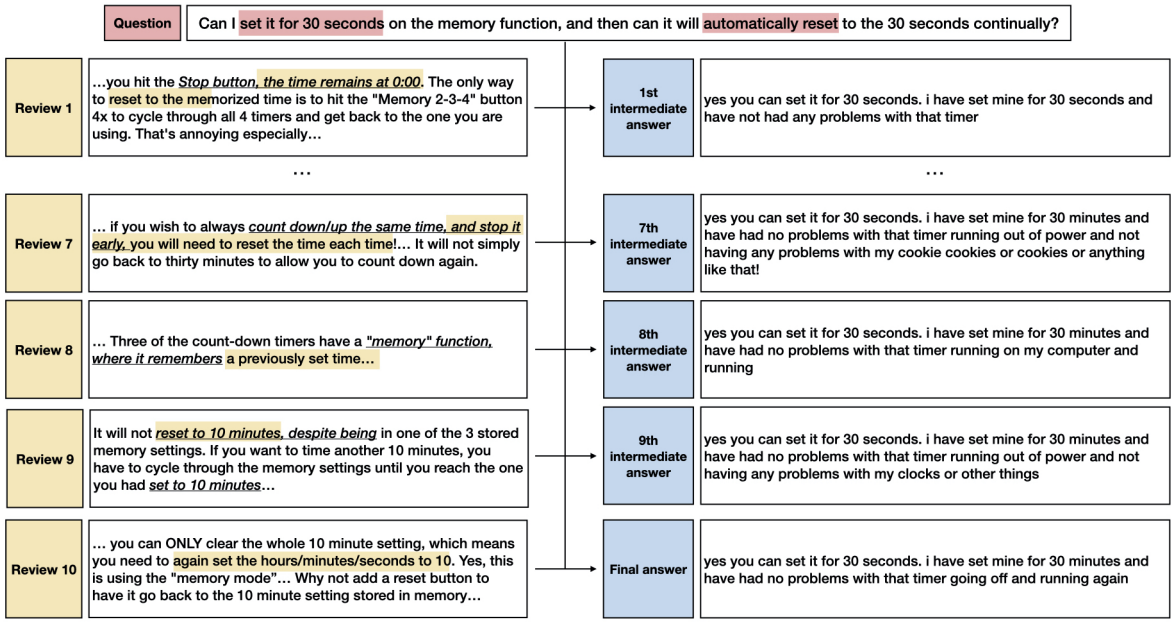

Figure 3: A breakdown of CHIME's generative process. The example question (the top box), the paired reviews (left panel), and the intermediate answers (right panel) after gradually reading the corresponding reviews. The major points of the question and reviews are highlighted with colors, and the *italic text* marked with underline is the content most concerned by the forget gate. Given new reviews, the very first generated simple answer becomes complicated and full of noise, but finally converges to the most prominent opinions and facts relevant to the question.

Figure 3 shows a breakdown of CHIME's generative process. The question-related content highlighted with colors is highly likely the major concerning part that the forget gate believes to memorise. The intermediate answers reported show that CHIME has locked the answer to the first sub-question from the beginning. But for the second sub-question, as the 8th answer shows, CHIME was also misled by other unimportant information. The final answer is eventually a synthesis of the prominent opinions encountered, summarised in a few concise phrases.

5 Conclusions

In this paper, we have proposed CHIME, a cross-passage hierarchical memory network for multi-passage generative review QA. It is built on the XLNet generator (Yang et al., 2019) by adding a memory module consisting of a *context* and a *answer memory* which guarantees a more accurate refining process for cross-passage evidence collection and answer generation. The sequential process adopted in CHIME makes it possible to elaborate longer text passages and some straightforward interpretability. We have assessed experimentally a significant quality improvement using different state-of-the-art metrics to measure the lexical and semantic coherence of the generated text. We plan to further extend CHIME to model with multiple ground truth simultaneously and leverage the available product attributes.

6 Acknowledgements

We would like to thank Mansi Gupta and her co-authors for sharing the test set of AmazonQA (Gupta et al., 2019) with us. This work was funded in part by EPSRC (grant no. EP/T017112/1), EU-H2020 (grant no. 794196), the Natural Science Foundation of China (61976066), and the Fundamental Research Fund for the Central Universities (2019GA35).

References

Yu Chen, Lingfei Wu, and Mohammed J Zaki. 2019. Bidirectional attentive memory networks for question answering over knowledge bases. *arXiv preprint arXiv:1903.02188*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.

Chuang Fan, Qinghong Gao, Jiachen Du, Lin Gui, Ruifeng Xu, and Kam-Fai Wong. 2018. Convolution-based memory network for aspect-based sentiment analysis. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1161–1164.

Jinlan Fu, Yi Li, Qi Zhang, Qinzhuo Wu, Renfeng Ma, Xuanjing Huang, and Yu-Gang Jiang. 2020. Recurrent memory reasoning network for expert finding in community question answering. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 187–195.

Shen Gao, Zhaochun Ren, Yihong Zhao, Dongyan Zhao, Dawei Yin, and Rui Yan. 2019. Product-aware answer generation in e-commerce question-answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 429–437.

Lin Gui, Jiannan Hu, Yulan He, Ruifeng Xu, Qin Lu, and Jiachen Du. 2017. A question answering approach to emotion cause extraction. *arXiv preprint arXiv:1708.05482*.

Mansi Gupta, Nitish Kulkarni, Raghuveer Chanda, Anirudha Rayasam, and Zachary C Lipton. 2019. Amazonqa: a review-based question answering task. *arXiv preprint arXiv:1908.04364*.

John T. Guthrie and Peter Mosenthal. 1987. Literacy as multidimensional: Locating information and reading comprehension. *Educational Psychologist*, 22(3-4):279–297.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2018. Abstractive summarization of reddit posts with multi-level memory networks. *arXiv preprint arXiv:1811.00783*.

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 *International conference on machine learning*, pages 1378–1387.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Sameen Maruf and Gholamreza Haffari. 2017. Document context neural machine translation with memory networks. *arXiv preprint arXiv:1711.03688*.

Julian McAuley and Alex Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web*, pages 625–635.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

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 Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne, editors, *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*.

Kyosuke Nishida, Itsumi Saito, Kosuke Nishida, Kazutoshi Shinoda, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2019. Multi-style generative reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2273–2284, Florence, Italy, July. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Mengting Wan and Julian McAuley. 2016. Modeling ambiguity, subjectivity, and diverging viewpoints in opinion question answering systems. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 489–498. IEEE.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, July.

Yizhong Wang, Kai Liu, Jing Liu, Wei He, Yajuan Lyu, Hua Wu, Sujian Li, and Haifeng Wang. 2018. Multi-passage machine reading comprehension with cross-passage answer verification. *arXiv preprint arXiv:1805.02220*.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage bert: A globally normalized bert model for open-domain question answering. *arXiv preprint arXiv:1908.08167*.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2019a. Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*.

Kun Xu, Yuxuan Lai, Yansong Feng, and Zhiguo Wang. 2019b. Enhancing key-value memory neural networks for knowledge based question answering. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2937–2947.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Qian Yu and Wai Lam. 2018. Review-aware answer prediction for product-related questions incorporating aspects. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 691–699.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yukun Zheng, Jiaxin Mao, Yiqun Liu, Zixin Ye, and Shaoping Ma. 2019. Human behavior inspired machine reading comprehension. In *the 42nd International ACM SIGIR Conference*.

Appendix A More Example Outputs Generated by Various Models

(a) Question: does this come with the straps?/
Answer 1: no it doesn't sadly.
Answer 2: no. you have to buy separate.
Answer 3: it comes with the white cotton rope ending in knots typical of mayan hammocks but no tree hugger straps or hammock frame.
Answer 4: no it does not come with the straps or rope. i advise you to purchase parachord with it.
XLNet + Summary: yes, it comes with the straps and the hammock is included with it! it comes in two pairs of harnesses that are attached to each other, so you can use them on your porch or your bed, or just use them on a bed or bed
XLNet + V-Net: yes it comes with the straps and the sizing is very good and very easy to do!
CHIME: yes it does. i love this hammock and love the quality of the straps

---

(b) Question: has anyone tried baking with this?
Answer 1: yes you can! designer whey also has recipes that you can make, like cookies, waffles or pancakes and muffins!
XLNet + Summary: i have never tried it with my own breads but i have had a few good ones and they were great!
XLNet + V-Net: i haven't tried baking with it, but i've been using this for a couple of weeks now and
CHIME: i haven't tried baking with this, but the flavor is very good and the flavor is very nice and good!

(c) Question: can you put coals on the lid, like you can with some lodge models?
Answer 1: yes you can,
Answer 2: yes. i've done it many times. make sure you get one of those lid-lifters though so you don't pour ash or coals into the pot.
Answer 3: yes, this dutch oven is specifically designed with a tall lip to cup the coals on top of the lid to allow for baking. this has worked great for me in making cobbler and biscuits.
Answer 4: never heard of coals on the lid, sorry. it makes a great chili over an open pit bbq. we hang it over a pit in the winter out back and make stews, chili's and pot luck. makes for a fun time especially in a snow storm!
Answer 5: shouldn't be a problem. lid fits with good seal so no coals or ash should spill inside.
XLNet + Summary: yes, you can. i don't know if it is possible to put coals on the top of the lid and then place them on top of that
XLNet + V-Net: yes, you can put coals on the lid, but i don't think you can put coals inside the top, so i don've had any problems with this oven and have not tried any of these models with any problem with mine
CHIME: yes, you can put coals on the lid, but i don't think you can put any coal in this one, as the lid is not designed for that type of use

---

(d) Question: in what stores are these sold?
Answer 1: target and target.com is where i've bought 3 pairs over time.
Answer 2: different sizes different stores.
XLNet + Summary: i don't know. i have never used them in my life, but they are very good for the foot, but it is not as good as the ones that are sold at stores
XLNet + V-Net: i don't know. i've been using them for a couple of years and it works great! they are very good and very good!
CHIME: amazon has them in their store. i have had them for over 2 years and have never had any problems with it working for my feet, especially when

Table A1: Example outputs to compare the quality of CHIME answers and XLNets answers.

(e) Question: is this set compatible with the new canon powershot sx50 hs?
Answer 1: yes james, as long as you decide what mm size you want to use! i prefered the 58mm adapter ring, this info is available in the canon online operaters manual! just make sure you specify the camera number (sx50 hs) when ordering any after market devices!! ps: you may already have 52mm (or other) lenses from another camera,
Answer 2: yes
Answer 3: i don't know as i have the sx40. as far as the sx40 i can say that everything fits and works with the camera very nice. i've now used everything. the lens cover is much much better than the one that came with the camera. the little case that houses the lenses is very nice. once you screw the adapter ring onto the end
Answer 4: i bought it for my canon sx40 and it works perfectly, but i'm not sure if the sx50 lens diameter is larger. this set does come with an adapter ring that fit snugly on my canon sx40 which allows placement of the filters and you can also stack the filters and/or add the lens hood. i mainly just use the lens hood
Answer 5: i just recieved this in the mail and the adapter does not fit my sx50??
Answer 6: hi everyone! the sx40 and sx50 have very similar dimensions and many people were able to make this ring fit on the sx50, however please know that there is an adapter ring designed specifically for use with the sx50. you can purchase this exact kit with the sx50 ring by following this link essential accessory kit for canon powershot sx
XLNet + Summary: yes, it is compatible with the new canon powershot and the new canon power shot camera, but it is compatible with all other cameras that have a different camera and
XLNet + V-Net: yes it is compatible with the sx40 sx20
CHIME: yes, it is compatible with the new canon sx50

(f) Question: does it work on an iphone 4 or less?
Answer 1: yes, it will work. it will not work with any higher versions only lower.
Answer 2: yes as the plugs are the same.. but to tell you the truth i wouldn't bother buying one if i had my time over. not long after i ordered my dock sonos released an uprade, "play from this phone" which basically meant you didn't have to have your computer on the listen to music..i only bought the dock for this very purpose
XLNet + Summary: yes, it does. i have a iphone 4 and it works fine with my 4 year old
XLNet + V-Net: yes it works on my iphone 4 and it works great! i have a sonos and have had no problems with the device running with my sono 5, but the cord is not very long, so if you are looking for an older model you should be able to find one that will do that
CHIME: yes it works on my iphone 4 and yes, works with my sonos

(g) Question: how is this mounted on a deck railing that is 1 1/4" wide?
Answer 1: the hanger just sits over the railing but it also has the side piece that a lot of other planters don't have. that extra piece rests against the railing at a ninety degree angle and keeps the planter from tipping forward. it's great...i love mine!
Answer 2: it has adjustable brackets that attach on each end
Answer 3: it is adjustable
Answer 4: it has adjustable brackets. it works on my pull fence quite well.
XLNet + Summary: it's a 2" wide. it's a 2 1/4" long, but i would say that this is the best way to mount the rack on any railing, so if you are looking for an adjustable bracket for your deck or balcony or window
XLNet + V-Net: i've used mine on a deck with no problems. i've had no problems on my deck railing, but the rail is not sturdy enough to hold up to that height of my house
CHIME: i have a deck railing that is 1 1/4" wide. i would recommend using a bracket to hold the box in place, but if you are going to hang your deck railing over

Table A2: Example outputs to compare the quality of CHIME answers and XLNets answers.