Answer-based Adversarial Training for Generating Clarification Questions

Sudha Rao[*]
Microsoft Research, Redmond
Sudha.Rao@microsoft.com

Hal Daumé III
University of Maryland, College Park
Microsoft Research, New York City
me@hal3.name

Abstract

We present an approach for generating clarification questions with the goal of eliciting new information that would make the given textual context more complete. We propose that modeling hypothetical answers (to clarification questions) as latent variables can guide our approach into generating more useful clarification questions. We develop a Generative Adversarial Network (GAN) where the generator is a sequence-to-sequence model and the discriminator is a utility function that models the value of updating the context with the answer to the clarification question. We evaluate on two datasets, using both automatic metrics and human judgments of usefulness, specificity and relevance, showing that our approach outperforms both a retrieval-based model and ablations that exclude the utility model and the adversarial training.

1 Introduction

A goal of natural language processing is to develop techniques that enable machines to process naturally occurring language. However, not all language is clear and, as humans, we may not always understand each other (Grice, 1975); in cases of gaps or mismatches in knowledge, we tend to ask questions (Graesser et al., 2008). In this work, we focus on the task of automatically generating clarification questions: questions that ask for information that is *missing* from a given linguistic context. Our clarification question generation model builds on the sequence-to-sequence approach that has proven effective for several language generation tasks (Sutskever et al., 2014; Serban et al., 2016; Yin et al., 2016; Du et al., 2017). Unfortunately, training a sequence-to-sequence model directly on (context, question) pairs yields questions that are highly generic[1], corroborating a common finding in dialog systems (Li et al., 2016b). Our goal is to be able to generate clarification questions that are useful *and* specific.

To achieve this, we begin with a recent observation of Rao and Daumé III (2018), who consider the task of question reranking: a good clarification question is the one whose answer has a high *utility*, which they define as the likelihood that this question would lead to an answer that will make the context more complete (§2.3). Inspired by this, we construct a model that first generates a question given a context, and then generates a hypothetical answer to that question. Given this (context, question, answer) triple, we train a utility calculator to estimate the usefulness of this question. We then show that this utility calculator can be generalized using ideas for generative adversarial networks (Goodfellow et al., 2014) for text (Yu et al., 2017), wherein the utility calculator plays the role of the "discriminator" and the question generator is the "generator" (§2.2), which we train using the MIXER algorithm (Ranzato et al., 2015). We evaluate our approach on two datasets: Amazon product descriptions (Figure 1) and Stack Exchange posts (Figure 2). Our two main contributions are:

1. An adversarial training approach for generating clarification questions that models the utility of updating a context with an answer to the clarification question. [2]
2. An empirical evaluation using both automatic metrics and human judgments to show that our adversarially trained model generates questions that are more *useful* and *specific to the context* than all the baseline models.

[*] This research performed when the author was still at University of Maryland, College Park.

[1] For instance, under home appliances, frequently asking "Is it made in China?" or "What are the dimensions?"

[2] Code and data: https://github.com/raosudha89/clarification_question_generation_pytorch

Proceedings of NAACL-HLT 2019, pages 143–155
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

Product title	T-fal Nonstick Cookware Set, 18 pieces, Red
Product description	Easy non-stick 18pc set includes every piece for your everyday meals. Exceptionally durable dishwasher safe cookware for easy clean up. Durable non-stick interior. Oven safe up to 350.F/177.C
Question	Are they induction compatible?
Answer	They are aluminium so the answer is NO.

Figure 1: Sample product description from Amazon paired with a clarification question and answer.

Title	Wifi keeps dropping on 5Ghz network
Post	Recently my wireless has been iffy at my university. I notice I am connected to a 5Ghz network, while I am connected to a 2.4Ghz everywhere else (where things work fine). Sometimes it reconnects, but I have to run 'sudo service network-manager restart'. Is it possible a kernel update caused this?
Question	what is the make of your wifi card ?
Answer	intel corporation wireless 7260 (rev 73)

Figure 2: Sample post from stackexchange.com paired with a clarification question and answer.

2 Training a Clarification Question Generator

Our goal is to build a model that, given a context, can generate an appropriate clarification question. Our dataset consists of (*context*, *question*, *answer*) triples where the *context* is an initial textual context, *question* is the clarification question that asks about some missing information in the context and *answer* is the answer to the clarification question (details in § 3.1). Representationally, our question generator is a standard sequence-to-sequence model with attention (§2.1). The learning problem is: how to train the sequence-to-sequence model to generate good clarification questions.

An overview of our training setup is shown in Figure 3. Given a context, our question generator, which is a sequence-to-sequence model, outputs a question. In order to evaluate the usefulness of this question, we then have a second sequence-to-sequence model called the "answer generator" that generates a hypothetical answer based on the context and the question (§ 2.5). This (context, generated question and generated answer) triple is fed into a UTILITY calculator, whose initial goal is to estimate the probability that this (question, answer) pair is useful in this context (§ 2.3). This UTILITY is treated as a reward, which is used to update the question generator using the MIXER (Ranzato et al., 2015) algorithm (§ 2.2). Finally, we reinterpret the answer-generator-plus-utility-calculator component as a *discriminator* for differentiating between (context, true question, generated answer) triples and (context, generated question, generated answer) triples , and optimize the generator for this adversarial objective using MIXER (§2.4).

2.1 Sequence-to-sequence Model for Question Generation

We use a standard attention based sequence-to-sequence model (Luong et al., 2015) for our question generator. Given an input sequence (context) $c = (c_1, c_2, ..., c_N)$, this model generates an output sequence (question) $q = (q_1, q_2, ..., q_T)$. The architecture of this model is an encoder-decoder with attention. The encoder is a recurrent neural network (RNN) operating over the input word embeddings to compute a source context representation $\tilde{c}$. The decoder uses this source representation to generate the target sequence one word at a time:

$$\begin{aligned} p(q|\tilde{c}) &= \prod_{t=1}^{T} p(q_t|q_1, q_2, ..., q_{t-1}, \tilde{c}_t) \\ &= \prod_{t=1}^{T} \textit{softmax}(W_s \tilde{h}_t) \quad ; \\ \text{where } \tilde{h}_t &= \tanh(W_c[\tilde{c}_t; h_t]) \end{aligned} \tag{1}$$

In Eq 1, $\tilde{h}_t$ is the attentional hidden state of the RNN at time t and W_s and W_c are parameters of the model.[3] The predicted token q_t is the token in the vocabulary that is assigned the highest probability using the softmax function. The standard training objective for sequence-to-sequence model is to maximize the log-likelihood of all (c, q) pairs in the training data D which is equivalent to minimizing the following loss,

$$L_{\text{mle}}(D) = - \sum_{(c,q)\in D} \sum_{t=1}^{T} \log p(q_t|q_1, ..., q_{t-1}, \tilde{c}_t) \tag{2}$$

[3] Details are in Appendix A.

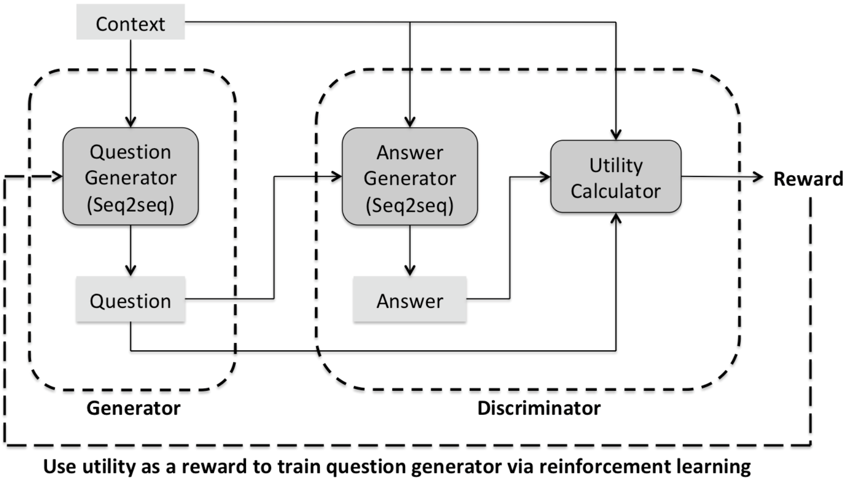
Context
Question Generator (Seq2seq)
Question
Answer Generator (Seq2seq)
Answer
Utility Calculator
Reward
Generator
Discriminator
Use utility as a reward to train question generator via reinforcement learning

Figure 3: Overview of our GAN-based clarification question generation model (refer preamble of §2)

2.2 Training the Generator to Optimize UTILITY

Training sequence-to-sequence models for the task of clarification question generation (with context as input and question as output) using maximum likelihood objective unfortunately leads to the generation of highly generic questions, such as *"What are the dimensions?"* when asking questions about home appliances. Recently, Rao and Daumé III (2018) observed that the usefulness of a question can be better measured as the *utility* that would be obtained if the context were updated with the answer to the proposed question. Following this observation, we first use a pretrained answer generator (§2.5) to generate an answer given a context and a question. We then use a pretrained UTILITY calculator (§ 2.3) to predict the likelihood that the generated answer would increase the utility of the context by adding useful information to it. Finally, we train our question generator to optimize this UTILITY based reward.

Similar to optimizing metrics like BLEU and ROUGE, this UTILITY calculator also operates on discrete text outputs, which makes optimization difficult due to non-differentiability. A successful recent approach dealing with the non-differentiability while also retaining some advantages of maximum likelihood training is the Mixed Incremental Cross-Entropy Reinforce (Ranzato et al., 2015) algorithm (MIXER). In MIXER, the overall loss L is differentiated as in REINFORCE (Williams, 1992):

$$\begin{aligned} L(\theta) &= -\mathbb{E}_{q^s \sim p_\theta} r(q^s) \quad ; \\ \nabla_\theta L(\theta) &= -\mathbb{E}_{q^s \sim p_\theta} r(q^s) \nabla_\theta \log p_\theta(q^s) \end{aligned} \tag{3}$$

where q^s is a random output sample according to the model p_θ and θ are the parameters of the network. The expected gradient is then approximated using a single sample $q^s = (q^s_1, q^s_2, ..., q^s_T)$ from the model distribution (p_θ). In REINFORCE, the policy is initialized randomly, which can cause long convergence times. To solve this, MIXER starts by optimizing maximum likelihood for the initial Δ time steps, and slowly shifts to optimizing the expected reward from Eq 3 for the remaining $(T - \Delta)$ time steps.

In our model, for the initial Δ time steps, we minimize L_{mle} and for the remaining steps, we minimize the following UTILITY-based loss:

$$L_{\text{max-utility}} = -(r(q^p) - r(q^b)) \sum_{t=1}^{T} \log p(q_t | q_1, ..., q_{t-1}, \tilde{c}_t) \tag{4}$$

where $r(q^p)$ is the UTILITY based reward on the predicted question and $r(q^b)$ is a baseline reward introduced to reduce the high variance otherwise observed when using REINFORCE. To estimate this baseline reward, we take the idea from the self-critical training approach Rennie et al. (2017) where the baseline is estimated using the reward obtained by the current model under greedy decoding during test time. We find that this approach for baseline estimation stabilizes our model better than the approach used in MIXER.

2.3 Estimating UTILITY from Data

Given a (context, question, answer) triple, Rao and Daumé III (2018) introduce a utility calculator UTILITY(c, q, a) to calculate the value of updating a context c with the answer a to a clarification question q. They use the utility calculator to estimate the probability that an *answer* would be a meaningful addition to a context. They treat this as a binary classification problem where the positive instances are the true (context, question, answer) triples in the dataset whereas the negative instances are contexts paired with a random (question, answer) from the dataset. Following Rao and Daumé III (2018), we model our UTILITY calculator by first embedding the words in c and then using an LSTM (long-short term memory) (Hochreiter and Schmidhuber, 1997) to generate a neural representation $\bar{c}$ of the context by averaging the output of each of the hidden states. Similarly, we obtain neural representations $\bar{q}$ and $\bar{a}$ of q and a respectively using a question and an answer LSTM models. Finally, we use a feed forward neural network $F_{\text{UTILITY}}(\bar{c}, \bar{q}, \bar{a})$ to predict the usefulness of the question.

2.4 UTILITY GAN for Clarification Question Generation

The UTILITY calculator trained on true vs random samples from real data (as described in the previous section) can be a weak reward signal for questions generated by a model due to the large discrepancy between the true data and the model's outputs. In order to strengthen the reward signal, we reinterpret the UTILITY calculator (coupled with the answer generator) as a discriminator in an adversarial learning setting. That is, instead of taking the UTILITY calculator to be a fixed model that outputs the expected quality of a (question, answer) pair, we additionally optimize it to distinguish between true (question, answer) pairs and model-generated ones. This reinterpretation turns our model into a form of a generative adversarial network (GAN) (Goodfellow et al., 2014).

GAN is a training procedure for "generative" models that can be interpreted as a game between a generator and a discriminator. The generator is a model $g \in \mathcal{G}$ that produces outputs (in our case, questions). The discriminator is another model $d \in \mathcal{D}$ that attempts to classify between true outputs and model-generated outputs. The goal of the generator is to generate data such that it can fool the discriminator; the goal of the discriminator is to be able to successfully distinguish between real and generated data. In the process of trying to fool the discriminator, the generator produces data that is as close as possible to the real data distribution. Generically, the GAN objective is:

$$L_{\text{GAN}}(\mathcal{D}, \mathcal{G}) = \max_{d \in \mathcal{D}} \min_{g \in \mathcal{G}} \mathbb{E}_{x \sim \hat{p}} \log d(x) + \mathbb{E}_{z \sim p_z} \log(1 - d(g(z))) \tag{5}$$

where x is sampled from the true data distribution $\hat{p}$, and z is sampled from a prior defined on input noise variables p_z.

Although GANs have been successfully used for image tasks, training GANs for text generation is challenging due to the discrete nature of outputs in text. The discrete outputs from the generator make it difficult to pass the gradient update from the discriminator to the generator. Recently, Yu et al. (2017) proposed a sequence GAN model for text generation to overcome this issue. They treat their generator as an agent and use the discriminator as a reward function to update the generative model using reinforcement learning techniques. Our GAN-based approach is inspired by this sequence GAN model with two main modifications: a) We use MIXER algorithm as our generator (§ 2.2) instead of a purely policy gradient approach; and b) We use UTILITY calculator (§ 2.3) as our discriminator instead of a convolutional neural network (CNN).

Theoretically, the discriminator should be trained using (context, true question, true answer) triples as positive instances and (context, generated question, generated answer) triples as the negative instances. However, we find that training a discriminator using such positive instances makes it very strong since the generator would have to not only generate real looking questions but also generate real looking answers to fool the discriminator. Since our main goal is question generation and since we use answers only as latent variables, we instead use (context, true question, *generated answer*) as our positive instances where we use the pretrained answer generator to get the *generated answer* for the true question. Formally, our objective function is:

$$L_{\text{GAN-U}}(\mathcal{U}, \mathcal{M}) = \max_{u \in \mathcal{U}} \min_{m \in \mathcal{M}} \mathbb{E}_{q \sim \hat{p}} \log u(c, q, \mathcal{A}(c, q)) + \mathbb{E}_{c \sim \hat{p}} \log(1 - u(c, m(c), \mathcal{A}(c, m(c)))) \tag{6}$$

where $\mathcal{U}$ is the UTILITY discriminator, $\mathcal{M}$ is the MIXER generator, $\hat{p}$ is our data of (context, question, answer) triples and $\mathcal{A}$ is the answer generator.

2.5 Pretraining

Question Generator. We pretrain our question generator using the sequence-to-sequence model

(§2.1) to maximize the log-likelihood of all (context, question) pairs in the training data. Parameters of this model are updated during adversarial training.

Answer Generator. We pretrain our answer generator using the sequence-to-sequence model (§2.1) to maximize the log-likelihood of all ([context+question], answer) pairs in the training data. Parameters of this model are kept fixed during the adversarial training.[4]

Discriminator. In our UTILITY GAN model (§ 2.4), the discriminator is trained to differentiate between true and generated questions. However, since we want to guide our UTILITY based discriminator to also differentiate between true ("good") and random ("bad") questions, we pretrain our discriminator in the same way we trained our UTILITY calculator. For positive instances, we use a context and its true question, answer from the training data and for negative instances, we use the same context but randomly sample a question from the training data (and use the answer paired with that random question).

3 Experimental Results

We base our experimental design on the following research questions:

1. Do generation models outperform simpler retrieval baselines?
2. Does optimizing the UTILITY reward improve over maximum likelihood training?
3. Does using adversarial training improve over optimizing the pretrained UTILITY?
4. How do the models perform when evaluated for nuances such as specificity & usefulness?

3.1 Datasets

We evaluate our model on two datasets.

Amazon. In this dataset, *context* is a product description on amazon.com combined with the product title, *question* is a clarification question asked to the product and *answer* is the seller's (or other users') reply to the question. To obtain these data triples, we combine the Amazon question-answering dataset (McAuley and Yang, 2016) with the Amazon reviews dataset (McAuley et al., 2015). We show results on the `Home & Kitchen` category of this dataset since it contains a large number of questions and is relatively easier for human-based evaluation. It consists of $19,119$ training, $2,435$ tune and $2,305$ test examples (product descriptions), with 3 to 10 questions (average: 7) per description.

Stack Exchange. In this dataset, *context* is a post on stackexchange.com combined with the title, *question* is a clarification question asked in the comments section of the post and *answer* is either the update made to the post in response to the question or the author's reply to the question in the comments section. Rao and Daumé III (2018) curated a dataset of $61,681$ training, $7,710$ tune and $7,709$ test such triples from three related subdomains on stackexchage.com (askubuntu, unix and superuser). Additionally, for 500 instances each from the tune and the test set, their dataset includes 1 to 6 other questions identified as valid questions by expert human annotators from a pool of candidate questions.

3.2 Baselines and Ablated Models

We compare three variants (ablations) of our proposed approach, together with an information retrieval baseline:

GAN-Utility is our full model which is a UTILITY calculator based GAN training (§2.4) including the UTILITY discriminator and the MIXER question generator.[5]

Max-Utility is our reinforcement learning baseline where the pretrained question generator model is further trained to optimize the UTILITY reward (§2.2) without the adversarial training.

MLE is the question generator model pretrained on context, question pairs using maximum likelihood objective (§2.1).

Lucene[6] is our information retrieval baseline similar to the Lucene baseline described in Rao and Daumé III (2018). Given a context in the test set, we use Lucene, which is a TF-IDF based document ranker, to retrieve top 10 contexts that are most similar to the given context in the train set. We randomly choose a question from the human written questions paired with these 10 contexts in the train set to construct our Lucene baseline[7].

[4] We leave the experimentation of updating parameters of answer generator during adversarial training to future work.

[5] Experimental details are in Appendix B.

[6] `https://lucene.apache.org/`

[7] For the Amazon dataset, we ignore questions asked to products of the same brand as the given product since Amazon replicates questions across same brand allowing the true question to be included in that set.

3.3 Evaluation Metrics

We evaluate initially with automated evaluation metrics, and then more substantially with crowdsourced human judgments.

3.3.1 Automatic Metrics

Diversity, which calculates the proportion of unique trigrams in the output to measure the diversity as commonly used to evaluate dialogue generation (Li et al., 2016b).

BLEU (Papineni et al., 2002) [8], which evaluates n-gram precision between the output and the references.

METEOR (Banerjee and Lavie, 2005), which is similar to BLEU but includes stemmed and synonym matches to measure similarity between the output and the references.

3.3.2 Human Judgements

We use Figure-Eight[9], a crowdsourcing platform, to collect human judgements. Each judgement[10] consists of showing the crowdworker a context and a generated question and asking them to evaluate the question along following axes:

Relevance: We ask *"Is the question on topic?"* and let workers choose from: Yes (1) and No (0)

Grammaticality: We ask *"Is the question grammatical?"* and let workers choose from: Yes (1) and No (0)

Seeking new information: We ask *"Does the question ask for new information currently not included in the description?"* and let workers choose from: Yes (1) and No (0)

Specificity: We ask *"How specific is the question?"* and let workers choose from:

4: Specific pretty much only to this product (or same product from different manufacturer)

3: Specific to this and other very similar products

2: Generic enough to be applicable to many other products of this type

1: Generic enough to be applicable to any product under Home and Kitchen

0: N/A (Not applicable) i.e. Question is not on topic OR is incomprehensible

Usefulness: We ask *"How useful is the question to a potential buyer (or a current user) of the product?"* and let workers choose from:

4: Useful enough to be included in the product description

3: Useful to a large number of potential buyers (or current users)

2: Useful to a small number of potential buyers (or current users)

1: Useful only to the person asking the question

0: N/A (Not applicable) i.e. Question is not on topic OR is incomprehensible OR is not seeking new information

Criteria	Agreement
Relevance	0.92
Grammaticality	0.92
Seeking new information	0.84
Usefulness	0.65
Specificity	0.72

Table 1: Inter-annotator agreement on the five criteria used in human-based evaluation.

3.3.3 Inter-annotator Agreement

Table 1 shows the inter-annotator agreement (reported by Figure-Eight as confidence[11]) on each of the above five criteria. Agreement on *Relevance*, *Grammaticality* and *Seeking new information* is high. This is not surprising given that these criteria are not very subjective. On the other hand, the agreement on usefulness and specificity is quite moderate since these judgments can be very subjective.

Since the inter-annotator agreement on the usefulness criteria was particularly low, in order to reduce the subjectivity involved in the fine grained annotation, we convert the range [0-4] to a more coarse binary range [0-1] by mapping the scores 4 and 3 to **1** and the scores 2, 1 and 0 to **0**.

3.4 Automatic Metric Results

Table 2 shows the results on the two datasets when evaluated according to automatic metrics.

In the Amazon dataset, GAN-Utility outperforms all ablations on DIVERSITY, suggesting that it produces more diverse outputs. Lucene, on the other hand, has the highest DIVERSITY since it consists of human written questions, which tend to be more diverse because they are much longer compared to model generated questions. This comes at the cost of lower match with the reference as visible in the BLEU and METEOR scores.

[8] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

[9] https://www.figure-eight.com

[10] We paid crowdworkers 5 cents per judgment and collected five judgments per question.

[11] https://success.figure-eight.com/hc/en-us/articles/201855939-How-to-Calculate-a-Confidence-Score

	Amazon			StackExchange		
Model	DIVERSITY	BLEU	METEOR	DIVERSITY	BLEU	METEOR
Reference	0.6934	—	—	0.7509	—	—
Lucene	0.6289	4.26	10.85	0.7453	1.63	7.96
MLE	0.1059	**17.02**	12.72	0.2183	3.49	8.49
Max-Utility	0.1214	16.77	12.69	**0.2508**	3.89	8.79
GAN-Utility	**0.1296**	15.20	**12.82**	0.2256	**4.26**	**8.99**

Table 2: DIVERSITY as measured by the proportion of unique trigrams in model outputs. Bigrams and unigrams follow similar trends. BLEU and METEOR scores using up to 10 references for the Amazon dataset and up to six references for the StackExchange dataset. Numbers in bold are the highest among the models. All results for Amazon are on the entire test set whereas for StackExchange they are on the 500 instances of the test set that have multiple references.

In terms of BLEU and METEOR, there is inconsistency. Although GAN-Utility outperforms all baselines according to METEOR, the fully ablated MLE model has a higher BLEU score. This is because BLEU score looks for exact n-gram matches and since MLE produces more generic outputs, it is much more likely that it will match one of 10 references compared to the specific/diverse outputs of GAN-Utility, since one of those ten is highly likely to itself be generic.

In the StackExchange dataset GAN-Utility outperforms all ablations on both BLEU and METEOR. Unlike in the Amazon dataset, MLE does not outperform GAN-Utility in BLEU. This is because the MLE outputs in this dataset are not as generic as in the amazon dataset due to the highly technical nature of contexts in StackExchange. As in the Amazon dataset, GAN-Utility outperforms MLE on DIVERSITY. Interestingly, the Max-Utility ablation achieves a higher DIVERSITY score than GAN-Utility. On manual analysis we find that Max-Utility produces longer outputs compared to GAN-Utility but at the cost of being less grammatical.

3.5 Human Judgements Analysis

Table 3 shows the numeric results of human-based evaluation performed on the reference and the system outputs on 300 random samples from the test set of the Amazon dataset.[12] All approaches produce relevant and grammatical questions. All models are all equally good at seeking new information, but are weaker than Lucene, which performs better at seeking new information but at the cost of much lower specificity and lower usefulness.

Our full model, GAN-Utility, performs significantly better at the usefulness criteria showing that the adversarial training approach generates more useful questions. Interestingly, all our models produce questions that are more useful than Lucene and Reference, largely because Lucene and Reference tend to ask questions that are more often useful only to the person asking the question, making them less useful for potential other buyers (see Figure 4). GAN-Utility also performs significantly better at generating questions that are more specific to the product (see details in Figure 5), which aligns with the higher DIVERSITY score obtained by GAN-Utility under automatic metric evaluation.

Table 5 contains example outputs from different models along with their usefulness and specificity scores. MLE generates questions such as *"is it waterproof?"* and *"what is the wattage?"*, which are applicable to many other products. Whereas our GAN-Utility model generates more specific question such as *"is this shower curtain mildew resistant?"*. Appendix C includes further analysis of system outputs on both Amazon and Stack Exchange datasets.

4 Related Work

Question Generation. Most previous work on question generation has been on generating reading comprehension style questions i.e. questions that ask about information present in a given text (Heilman, 2011; Rus et al., 2010, 2011; Duan et al., 2017). Our goal, on the other hand, is to generate questions whose answer cannot be found

[12] We could not ask crowdworkers evaluate the StackExchange data due to its highly technical nature.

Model	Relevant [0-1]	Grammatical [0-1]	New Info [0-1]	Useful [0-1]	Specific [0-4]
Reference	0.96	0.99	0.93	0.72	3.38
Lucene	**0.90**	**0.99**	**0.95**	0.68	2.87
MLE	**0.92**	**0.96**	0.85	0.91	3.05
Max-Utility	**0.93**	**0.96**	0.88	0.91	3.29
GAN-Utility	**0.94**	**0.96**	0.87	**0.96**	**3.52**

Table 3: Results of human judgments on model generated questions on 300 sample Home & Kitchen product descriptions. Numeric range corresponds to the options described in §3.3. The difference between the bold and the non-bold numbers is statistically significant with p <0.05. Reference is excluded in the significance calculation.

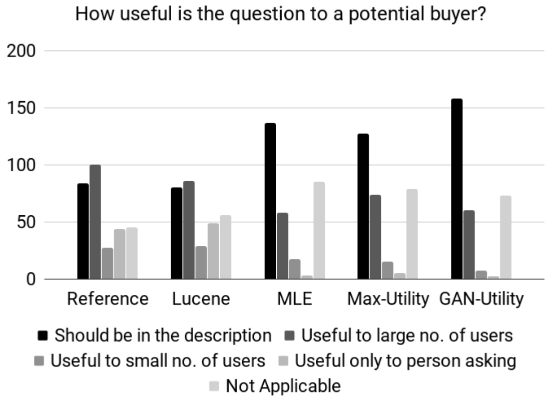

Figure 4: Human judgements on the usefulness criteria.

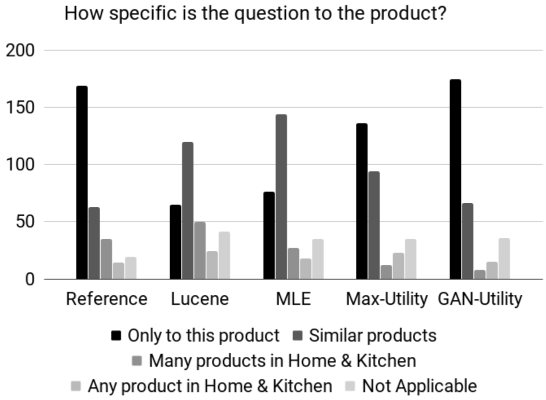

Figure 5: Human judgements on the specificity criteria.

in the given text. Outside reading comprehension questions, Liu et al. (2010) use templated questions to help authors write better related work sections whereas we generate questions to fill information gaps. Labutov et al. (2015) use crowdsourcing to generate question templates whereas we learn from naturally occurring questions. Mostafazadeh et al. (2016, 2017) generate natural and engaging questions, given an image (and some initial text). Whereas, we generate questions specifically for identifying missing information. Stoyanchev et al. (2014) generate clarification questions to resolve ambiguity caused by speech recognition failures during dialog, whereas we generate clarification questions to resolve ambiguity caused by missing information. The recent work most relevant to our work is by Rao and Daumé III (2018). They build a model which given a context and a set of candidate clarification questions, ranks them in a way that more useful clarification questions would be higher up in the ranking. In our work, we build on their ideas to propose a model that generates (instead of ranking) clarification questions given a context.

Neural Models and Adversarial Training for Text Generation. Neural network based models have had significant success at a variety of text generation tasks, including machine translation (Bahdanau et al., 2015; Luong et al., 2015), summarization (Nallapati et al., 2016), dialog (Bordes et al., 2016; Li et al., 2016a; Serban et al., 2017), textual style transfer (Jhamtani et al., 2017; Rao and Tetreault, 2018) and question answering (Yin et al., 2016; Serban et al., 2016). Our task is most similar to dialog, in which a wide variety of possible outputs are acceptable, and where lack of specificity in generated outputs is common. We addresses this challenge using an adversarial network approach (Goodfellow et al., 2014), a training procedure that can generate natural-looking outputs, which have been effective for natural image generation (Denton et al., 2015). Due to the challenges in optimizing over discrete output spaces like text, Yu et al. (2017) introduced a Seq(uence)GAN approach where they overcome this issue by using REINFORCE to optimize. Our GAN-Utility model is inspired by the SeqGAN model where we replace their policy gra-

Title	Raining Cats and Dogs Vinyl Bathroom **Shower Curtain**		
Product Description	This adorable shower curtain measures 70 by 72 inches and is sure to make a great gift!		
		Usefulness [0-4]	Specificity [0-4]
Reference	does the vinyl smells?	3	4
Lucene	other than home sweet home, what other sayings on the curtain?	2	4
MLE	is it waterproof ?	4	2
Max-Utility	is this shower curtain mildew ?	0	0
GAN-Utility	is this shower curtain mildew resistant ?	4	4
Title	PURSONIC HF200 Pedestal **Bladeless Fan & Humidifier** All-in-one		
Product Description	The first bladeless fan to incoporate a humidifier! This product operates solely as a fan, a humidifier or both simultaneously. Atomizing function via ultrasonic. 5.5L tank lasts up to 12 hours.		
		Usefulness [0-4]	Specificity [0-4]
Reference	i can not get the humidifier to work	1	2
Lucene	does it come with the vent kit	3	3
MLE	what is the wattage of this fan ?	4	2
Max-Utility	is this battery operated ?	3	2
GAN-Utility	does this fan have an automatic shut off ?	4	4

Table 4: Example outputs from each of the systems for two product descriptions along with the usefulness and the specificity score given by human annotators.

dient based generator with a MIXER model and their CNN based discriminator with our UTILITY calculator. Li et al. (2017) train an adversarial model similar to SeqGAN for generating next utterance in a dialog given a context. However, unlike our work, their discriminator is a binary classifier trained only to distinguish between human and machine generated utterances.

5 Conclusion

In this work, we describe a novel approach to the problem of clarification question generation. We use the observation of Rao and Daumé III (2018) that the usefulness of a clarification question can be measured by the value of updating a context with an answer to the question. We use a sequence-to-sequence model to generate a question given a context and a second sequence-to-sequence model to generate an answer given the context and the question. Given the (context, generated question, generated answer) triple, we calculate the utility of this triple and use it as a reward to retrain the question generator using reinforcement learning based MIXER model. Further, to improve upon the utility calculator, we reinterpret it as a discriminator in an adversarial setting and train both the utility calculator and the MIXER model in a minimax fashion. We find that our adversarial training approach produces more useful and specific questions compared to both a model trained using maximum likelihood objective and a model trained using utility reward based reinforcement learning.

There are several avenues of future work. Following Mostafazadeh et al. (2016), we could combine text input with image input in the Amazon dataset (McAuley and Yang, 2016) to generate more relevant and useful questions. One significant research challenge in the space of free text generation problems when the set of possible outputs is large, is that of automatic evaluation (Lowe et al., 2016): in our results we saw some correlation between human judgments and automatic metrics, but not enough to trust the automatic metrics completely. Lastly, we hope to integrate such a question generation model into a real world platform like StackExchange or Amazon to understand the real utility of such models and to unearth additional research questions.

Acknowledgments

We thank the three anonymous reviewers for their helpful comments and suggestions. We also thank the members of the Computational Linguistics and Information Processing (CLIP) lab at University of Maryland for helpful discussions. This work was supported by NSF grant IIS-1618193. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the view of the sponsors.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

Emily L Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. Deep generative image models using a Laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems*, pages 1486–1494.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1342–1352.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680.

Art Graesser, Vasile Rus, and Zhiqiang Cai. 2008. Question classification schemes. In *Proc. of the Workshop on Question Generation*.

H Paul Grice. 1975. Logic and conversation. *1975*, pages 41–58.

Michael Heilman. 2011. *Automatic factual question generation from text*. Ph.D. thesis, Carnegie Mellon University.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 889–898.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 994–1003.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016b. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202.

Jiwei Li, Will Monroe, Tianlin Shi, Sėbastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 366–376. Association for Computational Linguistics.

Ryan Lowe, Iulian V. Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. On the evaluation of dialogue systems with next utterance classification. In *SIGDIAL*.

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

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM.

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. International World Wide Web Conferences Steering Committee.

Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios Spithourakis, and Lucy Vanderwende. 2017. Image-grounded conversations: Multimodal context for natural question and response generation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 462–472.

Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating natural questions about an image. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1802–1813.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1.

Sudha Rao and Joel Tetreault. 2018. Dear Sir or Madam, May I introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer. In *HLT-NAACL*. The Association for Computational Linguistics.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *CVPR*, volume 1, page 3.

Vasile Rus, Paul Piwek, Svetlana Stoyanchev, Brendan Wyse, Mihai Lintean, and Cristian Moldovan. 2011. Question generation shared task and evaluation challenge: Status report. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 318–320. Association for Computational Linguistics.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The first question generation shared task evaluation challenge. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 251–257. Association for Computational Linguistics.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, volume 16, pages 3776–3784.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI*, pages 3295–3301.

Svetlana Stoyanchev, Alex Liu, and Julia Hirschberg. 2014. Towards natural clarification questions in dialogue systems. In *AISB symposium on questions, discourse and dialogue*, volume 20.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

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

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural generative question answering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2972–2978. AAAI Press.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *arxiv*.

A Sequence-to-sequence model details

In this section, we describe some of the details of the attention based sequence-to-sequence model introduced in Section 2.1 of the main paper. In equation 1, $\tilde{h}_t$ is the attentional hidden state of the RNN at time t obtained by concatenating the target hidden state h_t and the source-side context vector $\tilde{c}_t$, and W_s is a linear transformation that maps h_t to an output vocabulary-sized vector. Each attentional hidden state $\tilde{h}_t$ depends on a distinct input context vector $\tilde{c}_t$ computed using a global attention mechanism over the input hidden states as:

$$\tilde{c}_t = \sum_{n=1}^{N} a_{nt} h_n \tag{7}$$

$$a_{nt} = \text{align}(h_n, h_t) \tag{8}$$

$$= \exp\left[h_t^T W_a h_n\right] \Big/ \sum_{n'} \exp\left[h_t^T W_a h_{n'}\right] \tag{9}$$

The attention weights a_{nt} is calculated based on the alignment score between the source hidden state h_n and the current target hidden state h_t.

B Experimental Details

In this section, we describe the details of our experimental setup.

We tokenize and lowercase all inputs (context, question and answers). We set the max length of context to be 100, question to be 20 and answer to be 20. We find that increasing the length of contexts (to 150 or 200) of question/ answer (to 40) yields similar results according to automatic metrics with increased experimentation time.

Our sequence-to-sequence model (Section 2.1) operates on word embeddings which are pretrained on in domain data using Glove (Pennington et al., 2014). As frequently used in previous work on neural network modeling, we use an embeddings of size 200 and a vocabulary with cut off frequency set to 10. During train time, we use teacher forcing (Williams and Zipser, 1989). During test time, we use beam search decoding with beam size 5.

We use a hidden layer of size two for both the encoder and decoder recurrent neural network models with size of hidden unit set to 100. We use a dropout of 0.5 and learning ratio of 0.0001. In the MIXER model, we start with $\Delta = T$ and decrease it by 2 for every epoch (we found decreasing Δ to 0 is ineffective for our task, hence we stop at 2).

C Analysis of System Outputs

C.1 Amazon Dataset

First half of Table 5 shows the system generated questions for three product descriptions in the Amazon dataset.

In the first example, the product is a shower curtain. The Reference question is specific and highly useful. Lucene, on the other hand, picks a moderately specific ("how to clean it?") but useful question. MLE model generates a generic but useful "is it waterproof?". Max-Utility generates comparatively a much longer question but in doing so loses out on relevance. This behavior of generating two unrelated sentences is observed quite a few times in both Max-Utility and GAN-Utility models. This suggests that these models, in trying to be very specific, end up losing out on relevance. In the same example, GAN-Utility also generates a fairly long question which, although awkwardly phrase, is quite specific and useful.

In the second example, the product is a Duvet Cover Set. Both Reference and Lucene questions here are examples of questions that are pretty much useful only to the person asking the question. We find many such questions in both Reference and Lucene outputs which is the main reason for the comparatively lower usefulness scores for their outputs. All three of our models generate irrelevant questions since the product description explicitly says that the set is full size.

In the last example, the product is a set of mopping clothes. Reference question is quite specific but has low usefulness. Lucene picks an irrelevant question. MLE and Max-Utility generate highly specific and useful questions. GAN-Utility generates an ungrammatical question by repeating the last word many times. We observe this behavior quite a few times in the outputs of both Max-Utility and GAN-Utility models suggesting that our sequence-to-sequence models are not very good at maintaining long range dependencies.

C.2 Stack Exchange Dataset

Second half of Table 5 shows the system generated questions for three posts from the Stack Exchange dataset.

The first example is of a post where someone describes their issue of not being able to recover from their boot. Reference and Lucene questions are useful. MLE generates a generic question that is not very useful. Max-Utility generates a useful question but has slight ungrammaticality in it. GAN-Utility, on the other hand, generates a specific and an useful question.

In the second example, again Reference and Lucene questions are useful. MLE generates a generic question. Max-Utility and GAN-Utility both generate fairly specific question but contain unknown tokens. The Stack Exchange dataset contains several technical terms leading to a long tail in the vocabulary. Owing to this, we find that both Max-Utility and GAN-Utility models generate many instances of questions with unknown tokens.

In the third example, the Reference question is very generic. Lucene asks a relevant question. MLE again generates a generic question. Both Max-Utility and GAN-Utility generate specific and relevant questions.

	Amazon Dataset		
Title	Extra Wide **Shower Curtain** for a Clawfoot Tub/opaque with Magnets		
Description	Approximately 180” Wide 70” Long. Extra Wide to keep the water and heat in. Opaque in color to let the light in. Can be used as a liner or the curtain itself.		
		Usefulness [0-4]	Specificity [0-4]
Reference	does this curtain come with hooks?	4	4
Lucene	how to clean it ? i had it over 10 months now . and some mildew grow on it	3	3
MLE	is it waterproof ?	1	2
Max-Utility	what is the diameter of the bottom of the shower ? i have a kohler shower curtain .	0	0
GAN-Utility	does this curtain have suction feet on the bottom of the shower curtain ?	3	4
Title	Chezmoi Collection 7-piece Chic Ruched White **Duvet Cover Set**, Full Size (with Throw Pillows)		
Description	Create a world of romance with the elegantly-styled, ruching,and luxurious all white duvet cover set.		
		Usefulness [0-4]	Specificity [0-4]
Reference	How long will it take to ship this to my pin code?	1	1
Lucene	and can you use the duvet as is ? if not what shall i purchase to put in it for winter or fall ?	1	4
MLE	what are the dimensions of the king size ?	0	0
Max-Utility	what are the dimensions of the king size ?	0	0
GAN-Utility	does the king size come with a duvet cover or do you have to buy a king duvet ?	0	0
Title	Microfiber 3-Pack, **Pro-Clean Mopping Cloths** for Braava Floor Mopping Robot		
Description	Braavas textured Pro-Clean microfiber mopping cloths remove dirt and hair from your floors. The cloths can be washed and used hundreds of times. They are compatible with all Braava models, including the Pro-Clean Reservoir Pad. Each cloth is easy to attach and remove from the magnetic cleaning pad.		
		Usefulness [0-4]	Specificity [0-4]
Reference	do i have to use a new cloth every time i want to clean my floor? $5/$6 seems expensive per clean	2	4
Lucene	do they remove pet odor ?	0	0
MLE	will these work with the scooba ?	3	3
Max-Utility	do these cloths work on hardwood floors ?	3	4
GAN-Utility	will this work with the scooba mop mop mop mop mop mop mop	0	0
	StackExchange Dataset		
Title	how can i recover my boot ?		
Post	since last week i am trying to recover my boot . after the last update for ubuntu 12.04 i lost it and nobody could help me , i used boot-repair but there are problems with dependences , which can’t be fix . i will be very grateful if somebody could help me .		
Reference	what happens when you try to boot ?		
Lucene	can you get into bios ?		
MLE	how much ram do you have ?		
Max-Utility	do you have a swap partition partition ?		
GAN-Utility	what happens when you try to boot into safe mode ?		
Title	packages have unmet dependencies when trying to install anything		
Post	i ’m running ubuntu 14.04 lts . ive recently run into this problem with several applications , it seems to happen to anything i need to update or install . ive tried a lot of solutions and nothing seems to work.		
Reference	sudo dpkg -reconfigure all ? ?		
Lucene	if you use the graphical package manager , does n’t add the required packages automatically ?		
MLE	how long did you wait ?		
Max-Utility	can you post the output of ‘apt-cache policy UNK ?		
GAN-Utility	can you post a screenshot of the output of ‘sudo apt-get install UNK		
Title	full lubuntu installation on usb (uefi capable)		
Post	i want to do a full lubuntu installation on a usb stick that can be booted in uefi mode. i do not want persistent live usb but a full lubuntu installation and that can boot fromanyuefi-capable computer ...		
Reference	hello and welcome on askubuntu . could you please clarify what you want ?		
Lucene	so , ubuntu was installed to the pen drive ?		
MLE	which version of ubuntu ?		
Max-Utility	do you have a live cd or usb stick ?		
GAN-Utility	what is the model of the usb stick ?		

Table 5: Example outputs from each of the systems for three product descriptions from the Home & Kitchen category of the Amazon dataset and for three posts of StackExchange dataset.