Diverse and Specific Clarification Question Generation with Keywords

First Author Name,[1] **Second Author Name,** [2] **Third Author Name** [1]
[1] Affiliation 1
[2] Affiliation 2
firstAuthor@affiliation1.com, secondAuthor@affilation2.com, thirdAuthor@affiliation1.com

Abstract

Task-oriented writings like product descriptions on Amazon often suffer from the missing of important aspects. *Clarification question generation* (CQGen) can be a promising approach to help alleviate the problem. Unlike traditional QGen assuming the existence of answers in the context and generating questions accordingly, CQGen mimics user behaviors of asking for unstated information. Previous works assume generating one CQ per context, but we claim that generating a group of CQs can be more beneficial by covering wider information needs. We thus propose the task of *Diverse CQGen* and propose a new model named *KPCNet*, which generates CQs with Keyword Prediction and Conditioning, to deal with this challenge while improving specificity at the same time. Automatic and human evaluation on 2 datasets (`Home & Kitchen`, `Office`) showed that KPCNet can generate more specific questions and promote better group-level diversity than several competitive baselines.

1 Introduction

The development of the Internet has spawned a number of task-oriented writings, such as product descriptions on Amazon. However, since authors cannot always have a thorough understanding of consumers' need, their writings usually miss something deemed important by the audience. For example, an American author may assume his machine be used on 110V, and thus omit this in description. Customers from 220V-using countries would pay special attention to the voltage attribute stated in the description. On finding this absent from context, some customers may ask CQs like "What is the voltage of this machine?" in customer QA, while others would turn to other products immediately, an unfortunate loss to the seller.

Clarification question[1] generation (CQGen) is a promising approach that mimics the user engagement to help alleviate the problem. Authors may request for CQs before publishing their writings from somewhere like the hypothetical writing assistant we illustrated in Figure 1, and supplement missing information accordingly.

CQGen is a challenging task. First, it requires the question to be *specific* while not being *repetitive* to existing context.

[1]questions asking for what's missing from a given context

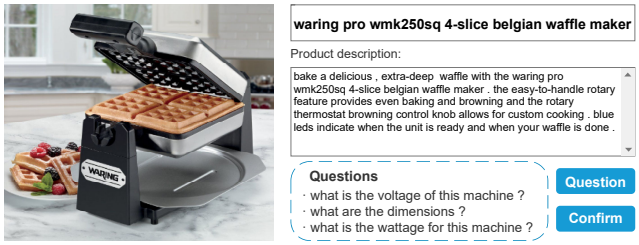

Figure 1: A hypothetical writing assistant generating CQs.

Questions with narrower application range are considered more specific. For example, the first question in Figure 1 is more specific than the second one because it should only be raised to electrical appliances. In contrast to the traditional QGen task which typically operates on the SQuAD 1.1 dataset (Rajpurkar et al. 2016) and derives the specificity from the knowledge of answer, CQGen doesn't expect the existence of answer in the context. Therefore, QGen algorithms which require answer and its position, either provided (Song et al. 2018; Sun et al. 2018) or can be extracted (Subramanian et al. 2018), are not applicable here. Vanilla seq2seq model has been shown to generate highly generic questions by Rao and Daumé III (2019). They then proposed GAN-Utility for this problem. However, their approach requires another QA component to generate answer from context and the generated question, which may not be reliable as the answers are inherently missing from context by definition. Consequently, this approach was shown to yield even worse result under some conditions (Cao, Rao, and Daumé III 2019). We thus totally eliminate the involvement of answers in our work, which will also enable the gathering of training data without answer.

Moreover, previous works on CQGen all assume generating one question per context. We claim that generating a group of diversity CQs (as is shown in Figure 1) can be more beneficial, because this allows the system to efficiently cover a variety of user needs at once, and become more robust to problems on any single generation. We name this novel task as **Diverse CQGen**. We seek algorithms that can deal with the task, and adopt a new group-level evaluation protocol to properly evaluate the system performance under this scenario.

To deal with the specificity challenge, we propose a novel model named Keyword Prediction and Conditioning Network (KPCNet). We observed that the main semantics of a question can be captured by its keywords. For example, the keyword of "What's the *dimension*?" is *dimension*, and the question can be comprehended even with a single word ("*dimension*?"). Keywords mainly consist of the product properties, and can allow for even more flexible and specific generation sometimes. For example, we can generate "Can you cook *rice* in this *cooker*?" with keywords "*cooker, rice*", while it will be unsuitable to use properties to exhaust all possible ingredients like *rice*. Therefore, the proposed KPCNet first predicts the probability for a keyword to appear in the generated CQ, then selects keywords from the predicted distribution, and finally conditions on them to generate questions. We can also partially control the generation by operating on the conditioned keywords, which can be utilized to avoid repetitive questions and further improve the quality.

To promote diversity, we explore several diverse generation approaches for this problem, including model-based *Mixture of Experts* (Shen et al. 2019) and decoding-based *Diverse Beam Search* (Vijayakumar et al. 2018). KPCNet's controllability through keywords also allows keywords-based approaches. We explore a novel use of classical clustering method on producing coherent keyword groups for keyword selection to generate correct, specific and diverse questions.

Individual and group-level evaluation showed that KPCNet is capable of producing more diverse and specific questions than competitive baselines, and can thus serve as a reliable backbone of the proposed writing assistant. Our contributions are:

1. We propose the task of *Diverse CQGen*, which requires generating a group of diverse CQs per context, to cover a wide range of information needs.
2. We propose KPCNet, which first predicts keywords for specificity, and then selects keywords as the condition for generation to allow partially controllable generation.
3. Based on the controllability of KPCNet, we propose several selection methods for conditioning keyword sets to promote diversity.
4. We evaluate KPCNet on 2 datasets (`Home & Kitchen` and `Office`) at both individual-level and group-level. A combination of automatic and human evaluation shows that KPCNet can achieve superior specificity in both settings and outperform strong baselines in diversity.

2 Preliminaries

2.1 Keyword-based Diverse CQGen

Given a textual *context* $x = (x_1, x_2, ..., x_{T_1})$, our aim is to generate a *clarification question* $y = (y_1, y_2, ..., y_{T_2})$, so that y asks for relevant but not repetitive information to x. In the setting of *Diverse CQGen*, we should generate a group of CQs for the same context such that they are semantically different from each other. In this work, we additionally consider *keywords* $Z = (z_1, z_2, ..., z_k)$ that are expected to capture the main semantic of y. Here we define keywords as lemmatized, non-stopping nouns, verbs and adjectives appearing in *questions*. Note that keywords are different from *answers*, and we don't assume the existence of answer in our approach. We extract a keyword dictionary $\mathcal{Z}$ of size C from the training set using this definition.

With keywords introduced, the marginal likelihood of a question are decomposed as:

$$\begin{aligned} p(y|x) &= \sum_{Z \subseteq \mathcal{Z}} p(y, Z|x) \\ &= \sum_{Z \subseteq \mathcal{Z}} p(y|x, Z) p(Z|x) \end{aligned} \tag{1}$$

2.2 Specificity

In this work, the specificity of a question is determined with the size of its applicable range. Question that can only be raised against one particular context is considered more specific than universal questions. Firstly, *relevance* is the basic requirement of specificity. Traditional MLE training may generate generic but not relevant question for higher likelihood. We hypothesize that the additional task of keyword prediction will help focus on relevant aspects. Besides, *specificity* of e-commerce questions are further promoted in 3 ways according to our observation : (1) Focusing on certain aspects, like the type, brand and attributes. (2) Mentioning components of a product, e.g. blade of a food grinder. (3) Describing a using experience specific to the product, such as cooking rice in a cooker. We hypothesize that many of them can be captured by keywords, with nouns and adjectives covering aspects and components, and verbs constituting the using experience.

2.3 Diversity

Diverse CQGen requires a group of questions to be generated about the same context, to cover various information needs as well as improve the robustness to problematic generations. This setting differs from some previous literatures (Wang et al. 2018; Rao and Daumé III 2019), where we generate only one response at a time, and *Diversity* is used to measure the expected variety of content and patterns among *all generated response*. We call it *global diversity*. Our setting is referred to as *local diversity*, measuring the diversity *within one usage*. This is also adopted by another line of literatures (Vijayakumar et al. 2018; Shen et al. 2019). If not specified, we mean *local diversity* by using *diversity*. Global diversity is also desired, as it increases the likelihood of the questions to be specific to various contexts.

To meet the diversity requirement as well as to promote specificity, we propose KPCNet below.

3 Model Description

In Equation 1, $p(Z|x)$ corresponds to the keyword prediction part, and $p(y|x, Z)$ refers to the keyword conditioned generation. Our model is thus divided into 2 logical parts. The whole model pipeline is illustrated in Figure 2.

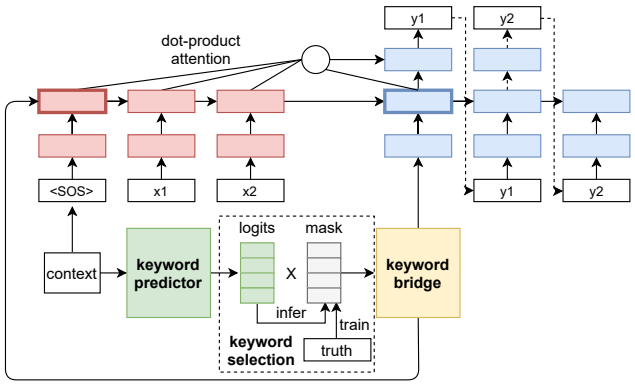

Figure 2: Illustration of KPCNet.

3.1 Keyword Prediction

For the *Keyword Predictor*, we assume the probability of each keyword z are independent from each other given context x, i.e. $p(Z|x) = \Pi_{z \subset Z} p(z|x)$, to simplify the modeling. We parameterize $p(z|x)$ with TextCNN (Kim 2014). The training loss is binary cross entropy over each keyword:

$$L_{pred} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} z_{n,c}^{t} log(p_{n,c}) \quad (2)$$

$z_{n,c}^{t}$ is a binary indicator shows if keyword c is among the ground truth keywords of the n_{th} sample, and $p_{n,c}$ is the predicted probability for c.

3.2 Keyword Conditioned Generation

The main structure of our generator is based on a standard sequence-to-sequence model (Luong, Pham, and Manning 2015). We will focus on our specific design to condition the generation on keywords.

Keyword Selection We take the unnormalized keyword logits $\hat{p} \in \mathbb{R}^C$ from the keyword predictor, and then we select a conditioning keyword set Z^s to mask out irrelevant dimensions to get a masked logits $\tilde{p} = [\hat{p}_1 z_1^s, \hat{p}_2 z_2^s, ..., \hat{p}_C z_C^s]$. This procedure allows us to control the generation with the selected keywords. Specific methods for this part will be discussed in Sec 3.3.

Keyword Bridge After getting the masked logits $\tilde{p}$, we pass them through a dropout layer, and then transform them to another distributed representation using a Multi-Layer Perceptron (MLP). They are then transformed into encoder features and decoder features with 2 MLPs respectively. The encoder feature will replace the hidden state of the first encoder step as memory to guide the generation via attention. The decoder feature will be fed as the input word embedding of the first decoder step to influence the generation.

3.3 Keyword Selection

At training, the ground truth keywords set Z^t is selected as Z^s, and the training objective is to maximize the log-likelihood of all questions given context x and keywords Z^t. This equals to minimize:

$$L_{mle} = -\frac{1}{N}\sum_{n=1}^{N} log(p(y_n|x_n, Z_n^t)) \quad (3)$$

At inference, we select Z^s from keyword predictor's predicted distribution as condition for generation. This process was done once at a time, and can be done several times to fully explore the diversity in $p(y|x)$ with different keyword sets. We come up with 3 methods for keyword selection:

Threshold We select all keywords whose predicted probability is above a threshold α as Z^s. If not specified, this is the default selection method at inference.

Sampling The threshold selection approach is deterministic and thus limited to one conditioning keyword set. We may encourage more diverse generation via diversifying the keyword set. An intuitive solution is to introduce randomness. Inspired by the Top-K (Fan, Lewis, and Dauphin 2018; Radford et al. 2019) and Top-p (nucleus) sampling (Holtzman et al. 2019), we also adopted a similar approach, sampling k keywords from softmax-normalized prediction distribution after Top-K, Top-p filtering.

Clustering Both of the threshold and sampling selection strategy run the risk of putting semantically uncoherent keywords together, which is the drawback of the independence assumption used by keyword predictor. For example, if *"voltage, machine, long, waffle"* are selected as the keywords for the waffle maker in Figure 1, we may generate an illogical question "what are the *voltage* of the *waffle*". To get more coherent keyword sets, we explore the use of clustering technique. For the above example, the keywords can form 2 semantic groups, which lead to "What are the *voltage* of the *machine*" and "How *long* does it take to cook *waffle*", respectively.

In practice, we first mine a keyword co-occurrence graph from the training set. We then take the Top-K likely keywords, and run Spectral clustering (Shi and Malik 2000) on the induced subgraph of them. The resulting g disjoint groups are then used as generation conditions respectively.

Keyword Controllability Probing One potential benefit that KPCNet brought is the controllability over generation by providing different conditioning keywords. To probe into this, we propose 2 approaches to operate on the keywords besides the 3 keywords selection methods. The operations are designed with hypotheses that will be tested with experiments.

Keyword Filtering Avoiding to ask existing information in the context is the basic requirement of CQ. However, none of existing methods in the literature proposed a specific approach to tackle this. In preliminary experiments of KPCNet, we found that some of the repetitive cases came with repetitive keywords. Therefore, we conjecture that we

Product	iliving organic buckwheat pillow with authentic japanese pillow cover, 14 by 20-inch, green
KPCNet	what is the **size** of this **pillow** case? (size, cover, pillow, wash, zipper)
+Filter	does this **pillow** have a **zipper**? (cover, pillow, wash, zipper)

Table 1: Example on the effect of keyword filtering. Predicted keywords for a question are shown in the parentheses below. "size" was filtered as it has already been covered in product description.

may alleviate the problem by filtering out such repetitive keywords. Table 1 provided a concrete example. This would be especially useful for iterative generation, as we will explicitly exclude repeating keywords if user triggers CQGen for the second time with some information vacancy already filled.

Here we use a simple matching method for keyword filtering. We first select a set of keywords that tends to lead to repetition. Then for each keyword in the set, we write a blacklist of words or patterns so that we filter the keyword if the pattern is matched. This process is currently done manually, so it doesn't scale. However, we find that a small set of frequent keywords is already enough to cover a relatively large number of repetitive cases and demonstrate the effect of this approach. We leave automatic repeating keyword detection and filtering for future works.

External Knowledge It is a common practice for e-commerce platforms to build knowledge graph to manage their products (Dong 2018; Luo et al. 2020). As a result, products are attached to highly related tags, concepts, or keywords in our terms. We hypothesize that such external knowledge may help the generation by directly providing high-quality keywords, or improving the keyword prediction. Nevertheless, since we don't have access to such knowledge, we simulate such scenario where we have higher quality keywords by directly feeding ground truth keywords to the model[KPCNet(truth)]. This establishes an upper bound to what extent can KPCNet be improved with knowledge.

3.4 Deduplication Postprocessing

All algorithms will more or less produce semantically similar questions in their initial generation group. Therefore, we will first generate more candidates than needed (say, produce 6 questions for 3 displaying slots), so that at least certain level of diversity can be guaranteed for the initial group. We then apply a simple, model-agnostic heuristic for deduplicating question selection. We first add the top generation into the current group, then we will iterative through the remaining questions. If the question's Jaccard similarity with any currently selected question is below 0.5, it will be added into the current group, otherwise it will be skipped.

4 Experiments

The research problems we want to tackle here are: (1) Can KPCNet generate more specific CQs? (2) To what extent can we control the generation of KPCNet by operating on the keywords? (3) How well can our proposed keyword selection methods promote local diversity, compared to existing diverse generation approaches?

4.1 Evaluation metrics

Most previous works on question generation (Jain, Zhang, and Schwing 2017; Hu et al. 2018; Rao and Daumé III 2019) adopts *Individual-level* evaluation protocol, where only the best generated question of a group is evaluated (thus also named *Oracle* metrics). Specially, for proper evaluation of the novel *Diverse CQGen* task, we need to evaluate the overall quality and diversity of CQ groups. We refer to this as *Group-level* evaluation. We adopt automatic metrics as well as human judgements on both level.

Automatic Metrics We use **Distinct-3** (DIVERSITY), **BLEU**[2](Papineni et al. 2002) and **METEOR** (Banerjee and Lavie 2005) for individual-level automatic evaluation.

For group-level evaluation, we adopt the evaluation protocol proposed by Shen et al. (2019) for diverse machine translation, and use **Pairwise-BLEU** and **Avg BLEU** as the evaluation metric.

Human Judgements For individual-level human judgements, we show the annotator one context and one generated question for each system (including reference). The system name is invisible to the annotator and the order is randomly shuffled. The selected candidate is the one that achieved the highest BLEU in the generation group.

We ask human to judge the **Grammaticality(G), Relevance(R), Seeking New Information(N) and Specificity(S)** of the questions. Also, noting that the system generations are also prone to make logical errors like improper repetition ("does the lid have a lid ?") or asking for relevant but not exactly the correct object (asking "what is the thickness of the bed ?" for a mattress), we further judge the **Logicality(L)** of the candidate.

For group-level human judgements, we run the deduplication procedure (Sec 3.4) to get 3 top questions for each system. And annotators are showed one context and the selected 3 questions for every group. The groups are also anonymized and shuffled.

For each question in a group, we score the same metrics as those for individual-level judgements. To evaluate the valid variety of each group produced by local generation diversity, we introduced an additional and important group-specific metric: **#Useful**. This is the number of useful questions after excluding problematic (ungrammatical, irrelevant, illogical, etc.) and semantically equivalent questions within a group. And we further calculate **#Redundant** as (the number of unproblematic questions - **#Useful**) to measure local redundancy.

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

Individual-level and group-level evaluation was conducted on the same set of 100 sample products for 8 systems and every group has 3 questions. They are distributed to 4 annotators so that each of the 2400 questions are annotated twice. We report inter-annotator agreement in Appendix.

4.2 Dataset

We evaluate our model on the `Home & Kitchen` category of the Amazon Dataset preprocessed by Rao and Daumé III (2019). We applied extra preprocessing on the raw data to remove noises in dataset (see Appendix). In this dataset, *context* is the product title concatenated with the product description, and *question* is the CQ asked by customers to the product. It consists of 19,119 training, 2,435 validation and 2,305 test examples (product descriptions), with 3 to 10 questions (average: 7) per description. The inherent diversity of questions in the dataset allows the proper evaluation of group-level generation diversity. We processed another category, `Office`, in a similar way. This is a much smaller dataset, consisting of 2,190 training, 285 validation and 256 test examples, with 3 to 10 questions (average: 6) per description. We will first analyze the results on `Home & Kitchen` in detail, then briefly discuss the results on `Office`.

4.3 Baselines

For individual-level generation, we compare KPCNet with the following models: **MLE** Vanilla seq2seq model trained on (context, question) pairs using maximum likelihood objective. **hMup** A representative of the family of mixture models proposed in Shen et al. (2019), which achieved a good balance of overall quality and diversity. Since we don't assume the availability of answers, we don't include traditional QGen methods and GAN-Utility (Rao and Daumé III 2019) in the comparison.

For a fair comparison, we control the encoder and decoder for all the above methods to have a similar 2-layer GRU (Cho et al. 2014) or LSTM (Hochreiter and Schmidhuber 1997) architecture and close amount of parameters.

For group-level generation, we compare across 3 categories of diverse generation methods:

Decoding based Classical beam search naturally produces different generation on each beam. Therefore, we evaluate the effect of beam search combined with MLE and KPCNet with threshold selection [KPCNet(beam)]. Recently, several decoding approaches (Ippolito et al. 2019) are proposed to further promote diversity in generation, among which *Diverse Beam Search*(Vijayakumar et al. 2018) and *Biased Sampling* like top-K, top-p sampling (Fan, Lewis, and Dauphin 2018; Holtzman et al. 2019) are representative methods. So we also evaluate KPCNet with them [KPCNet(divbeam), KPCNet(BSP)].

Model based hMup is designed for diversity at the model level. It provides a discrete latent variable called *expert* to control the generation. We thus take the top beam-searched candidate of each expert to form a generation group for evaluation.

Keywords based This is dedicated to KPCNet. We evaluate the *Sampling*[KPCNet(sample)] and *Clustering*[KPCNet(cluster)] methods for keyword selection. We also estimate the potential of KPCNet with knowledge (Sec 3.3) by providing the ground truth keyword set [KPCNet(truth)].

All systems using beam search have a beam size of 6, we also set number of experts for hMup to 6, and we use beam size of 6 with 3 diverse groups for *diverse beam search*. We select 2 keyword groups for KPCNet(sample) and KPCNet(cluster). To produce the final generation group for evaluation, outputs of all systems will go through the same deduplication postprocessing (Sec 3.4) to get 3 questions for each group.

4.4 `Home & Kitchen` Dataset Results

Individual-level Evaluation Table 2 shows the automatic evaluation results. KPCNet and hMup outperform MLE in METEOR but not in BLEU. We claim that it is due to the shorter and the safer generation of MLE, which is naturally favored by precision-based BLEU but not F-based METEOR. The average generation length is 5.957 for MLE, 8.231 for hMup, and 7.263 for KPCNet. KPCNet significantly outperform all the other baselines in Distinct-3 and METEOR, showing that KPCNet potentially promote higher global diversity and generation quality. We note that KPCNet(truth) has a great advantage over KPCNet, indicating the controllability of keywords and the potential of KPCNet to be further strengthened by improving the conditioned keyword set with other helpers like external knowledge (Sec 3.3).

	Distinct-3	BLEU	METEOR
ref	0.6934	-	-
MLE	0.0777	**18.13**	14.86
hMup	0.1111	17.76	15.40
KPCNet	**0.1530**	17.77	**16.18**
KPCNet(truth)	0.3738	23.63	19.38

Table 2: Individual-level automatic evaluation results on `Home & Kitchen` dataset.

	G	R	L	N	S
ref	0.98	1.00	1.00	0.94	2.68
MLE	**0.99**	0.92	**0.98**	**0.85**	1.45
hMup	**0.99**	0.92	0.86	0.81	1.81
KPCNet	**0.99**	**0.99**	0.95	0.80	1.81
KPCNet(filter)	**0.99**	**0.99**	0.94	**0.85**	**1.84**

Table 3: Individual-level human evaluation metrics on 100 sample products from `Home & Kitchen`. G/R/L/N/S stand for Grammatical, Relevance, Logicality, New Info and Specificity respectively.

	Relevant[0-1]	Logical[0-1]	New Info[0-1]	Specific[0-4]	#Useful[0-3]	#Redundant[0-2]	Avg Rank
ref	0.990	1.000	0.947	2.530	2.680	0.120	-
MLE	0.907	**0.943**	**0.863**	1.457	1.550	0.590	3.667
hMup	0.900	0.793	0.833	1.727	1.530	**0.130**	4.667
KPCNet(-filter)	**0.987**	0.870	0.830	1.757	1.280	0.750	4.500
KPCNet(beam)	**0.987**	0.853	**0.863**	1.793	1.330	0.750	3.667
KPCNet(divbeam)	0.963	0.780	0.860	1.760	1.480	0.310	4.167
KPCNet(sample)	0.963	0.837	0.850	**1.890**	1.500	0.450	3.500
KPCNet(cluster)	0.963	0.863	0.823	1.877	**1.760**	0.190	**3.000**

Table 4: Group-level human evaluation results on 100 sample products (300 questions each system) from `Home & Kitchen`. Grammatical is omitted as the results are similar to Table 3 where all systems performs well. *Avg Rank* is the average ranking among all 7 methods across the 6 metrics. We perform hypothesis test among KPCNet variants, and the difference between underlined and non-underlined numbers at each column is statistically significant with $p \leq 0.05$.

Table 3 shows the individual-level human evaluation results. We can see that all systems perform well in *Grammatical*, KPCNet significantly outperform other systems in *Relevant* and achieved the best *Specific*, while perform slightly worse in *Logical*. The superior *Relevant* proves our hypothesis that independently trained keyword predictor help focus on relevant keywords instead of irrelevant but generic words preferred by MLE (Sec 2.2). KPCNet(filter) gets a much higher *New Info* at the cost of only slight drop in *Logical*. This shows that the Keyword Filtering step (Sec 3.3) can truly utilize the controllability of keywords to help avoid repetition. Therefore, we by default incorporate the step with all the KPCNet variants in the next group-level evaluation stage while keep the plain KPCNet for comparison as KPCNet(-filter).

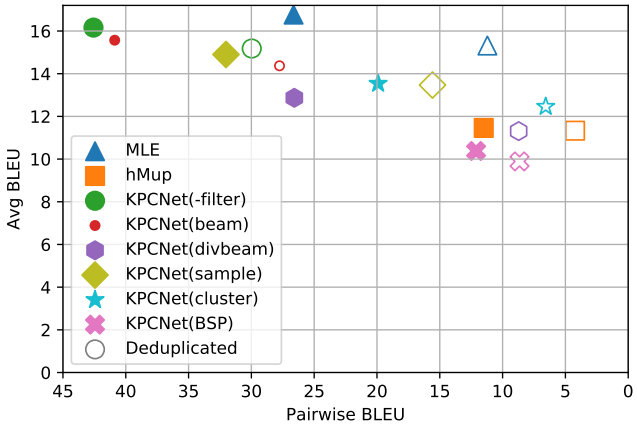

Figure 3: Group-level Automatic metrics on the whole test set of `Home & Kitchen`. The lower Pairwise BLEU, the more diverse for the generation group. Solid markers are the results for the top 3 candidates in the original group, while hollow markers measures the remaining 3 after deduplication. Points located near top-right are preferred as they achieve a good tradeoff between the 2 metrics.

Group-level Evaluation The group-level automatic evaluation metrics before and after deduplication for each system are shown in Figure 3. Original results are shown in solid markers. KPCNet(BSP) has the poorest Avg BLEU and we found the results very likely to be ungrammatical and illogical, and we thus omit it in later judgements. hMup has the highest local diversity while has the second poorest Avg BLEU. MLE has moderate level of local diversity and the highest Avg BLEU, and we found that Keyword Filtering slightly harmed Avg BLEU, which is against our intuition. But we later found Avg BLEU doesn't correlate well with most human judgements (see Appendix). Several diversity-promoting variants of KPCNet improved local diversity at the cost of Avg BLEU, among which KPCNet(cluster) achieved a best tradeoff between the two. Comparing the original and deduplicated results (hollow markers), we can see that our simple heuristic can effectively eliminate redundancy at the cost of slight degradation of Avg BLEU, as only nearly identical hypotheses with high BLEU are excluded.

Group-level human evaluation results are shown in Table 4. We can see that all KPCNet variants clearly outperform baselines in *Relevant* and *Specific* while have a competitive performance in *New info*. MLE rated best for *Logical* for its conservative generations (low *Specific*), and the questions tend to overlap with each other, as is reflected in high *#Redundant*. KPCNet(beam) has a even higher redundancy since its searching space is further limited by the conditioned keyword set. Diverse generation variants can help overcome this drawback. Especially, KPCNet(cluster) achieved the best *#Useful*, *Avg Rank*, and its performance on all metrics is among the best of KPCNet variants. This shows that the semantically-coherent keyword sets produced by clustering can effectively improve the generation diversity and quality of KPCNet.

Product	Novaform memory foam comfort curve pillow
KPCNet (cluster)	is this a **firm pillow**? (pillow, foam, sleep, firm) is this pillow good for **stomach sleepers**? (stomach, sleeper)
Product	full-sized headboard in solid wood
KPCNet (cluster)	what is the height of this **headboard** ? (bed frame headboard) does it have a **box spring** ? (mattress box spring)

Table 5: Example generation groups for KPCNet(cluster). Keywords in the parentheses.

	Grammatical[0-1]	Relevant[0-1]	Logical[0-1]	New Info[0-1]	Specific[0-4]	#Useful[0-3]	#Redundant[0-2]
ref	0.993	0.997	0.993	0.933	2.713	2.420	0.330
MLE	0.970	0.843	**0.883**	0.797	1.470	1.070	0.420
KPCNet	**0.993**	**0.940**	0.817	**0.803**	**1.903**	**1.470**	**0.190**

Table 6: Group-level human judgments on 100 samples from the `Office` dataset. KPCNet here uses keyword clustering.

Case Study Table 5 provided 2 example generation groups of KPCNet(cluster). For each group, the 6 predicted keywords captured specific aspects of the product. Then they are divided into 2 coherent groups (as they formed natural phrases such as "firm pillow" and "stomach sleeper") by clustering. Finally, the different conditioned keyword sets are reflected in the generation. In the first case, specific and diverse generations are successfully produced with precisely predicted keywords. We can see that the separation of keywords as controlling factors allows the novel use of classical clustering technique to help generate high-quality question groups by first producing coherent keyword sets. There are also bad cases like the second question in another group. The possible reason is that keyword predictor produced related but unsuitable keywords "box spring", which can be asked for a whole bed but not for headboard alone. This shows that predictor is the performance bottleneck of KPCNet. We provide an example of group-level human judgement results including all systems in Appendix.

4.5 `Office` Dataset Results

	Distinct-3	BLEU	METEOR
ref	0.7554	-	-
MLE	0.2033	**14.73**	13.81
hMup	0.1531	10.45	12.52
KPCNet	**0.3099**	13.84	**15.29**

Table 7: Individual-level automatic evaluation results on the `Office` dataset.

For brevity, we only show the individual-level automatic evaluation and group-level human judgement results. All the experimental settings are the same with the previous experiments, except that we apply no keyword filtering here.

Table 7 shows that KPCNet still outperforms MLE in Distinct-3 and METEOR, while falls behind at BLEU. Both the automatic metrics and our manual check indicate that hMup fails to give comparable performance at the small dataset, so we exclude it in group-level evaluation.

Table 6 showed that the performance of both models degraded here possibly due to the smaller data size. However, the observation is similar. KPCNet(cluster) outperforms MLE in most metrics especially at *Relevant*, *Specific* and *#Useful* despite a weakness at *Logical*. This shows that KPCNet(cluster) can consistently improve the diversity and specificity of the generation.

5 Related Work

Clarification Question Generation The concept of CQ can be naturally raised in a dialogue system for incomplete recognition (Stoyanchev, Liu, and Hirschberg 2014) or task-oriented slot-filling. The concept is then extended to IR to clarify ambiguous queries (Aliannejadi et al. 2019), and has been successfully put into practice (Zamani et al. 2020). Our work closely follows the research line of Rao and Daumé III (2018, 2019); Cao, Rao, and Daumé III (2019), which sets the problem in the task-oriented writing scenario. Rao and Daumé III (2018) first adopted a retrieval-then-rank approach, Rao and Daumé III (2019) then proposed a generation approach to train the model to maximize the utility of the hypothetical answer for the questions with GAN, to better promote specificity. Cao, Rao, and Daumé III (2019) propose to control the specificity by training on data with explicit indicator of specificity, but it requires additional specificity annotation. Towards the similar specificity goal, we adopted a different keyword-based approach. Parallel to our work, Kumar and Black (2020) proposed ClarQ, a large-scale and diverse dataset for CQGen derived from StackExchange.

Diverse Generation The demand for diverse generation exists in many other fields, and we've drawn inspirations from these literatures. For image captioning, we may use multiple descriptions for different focusing points of a scene. *Diverse Beam Search* (Vijayakumar et al. 2018) was proposed to broaden the searching space to catch such diversity by dividing groups in decoding and imposing repetition penalty between them. For machine translation, a context can be translated with different styles. Shen et al. (2019) thus proposed *Mixture of Expert* models including hMup to reflect various styles with a discrete latent variable (*expert*). And here for CQGen, diversity is required to cover various potentially missing aspects, so we come up with the idea to use keywords as a controlling variable like *expert* to promote diversity.

6 Conclusion

We propose the task of Diverse CQGen to request for various unstated aspects in a writing with a group of semantically different questions, in order to supplement those missing information before they cause negative results. We then propose KPCNet to deal with the novel task as well as improve the specificity of the questions. Human judgements showed that KPCNet is able to generate more specific questions and promote better group-level diversity. Oracle tests with ground truth keywords provided in keyword selection showed strong performance, indicating the great potential to be exploited from improving keyword prediction. Future works may include utilizing external knowledge to improve the keyword prediction, and solutions for the occasionally illogical generations.

References

Aliannejadi, M.; Zamani, H.; Crestani, F.; and Croft, W. B. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 475–484.

Banerjee, S.; and Lavie, A. 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*, 65–72.

Cao, Y. T.; Rao, S.; and Daumé III, H. 2019. Controlling the Specificity of Clarification Question Generation. In *Proceedings of the 2019 Workshop on Widening NLP*, 53–56.

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Dong, X. L. 2018. Challenges and innovations in building a product knowledge graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2869–2869.

Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 889–898. Melbourne, Australia: Association for Computational Linguistics. doi:10.18653/v1/P18-1082. URL https://www.aclweb.org/anthology/P18-1082.

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

Holtzman, A.; Buys, J.; Forbes, M.; and Choi, Y. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751* .

Hu, W.; Liu, B.; Ma, J.; Zhao, D.; and Yan, R. 2018. Aspect-based Question Generation .

Ippolito, D.; Kriz, R.; Sedoc, J.; Kustikova, M.; and Callison-Burch, C. 2019. Comparison of Diverse Decoding Methods from Conditional Language Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3752–3762.

Jain, U.; Zhang, Z.; and Schwing, A. G. 2017. Creativity: Generating diverse questions using variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6485–6494.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751. Doha, Qatar: Association for Computational Linguistics. doi:10.3115/v1/D14-1181. URL https://www.aclweb.org/anthology/D14-1181.

Kumar, V.; and Black, A. W. 2020. ClarQ: A large-scale and diverse dataset for Clarification Question Generation. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* doi:10.18653/v1/2020.acl-main.651. URL http://dx.doi.org/10.18653/v1/2020.acl-main.651.

Luo, X.; Liu, L.; Yang, Y.; Bo, L.; Cao, Y.; Wu, J.; Li, Q.; Yang, K.; and Zhu, K. Q. 2020. AliCoCo: Alibaba E-commerce Cognitive Concept Net. *arXiv preprint arXiv:2003.13230* .

Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1412–1421.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners .

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392.

Rao, S.; and Daumé III, H. 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)*, 2737–2746.

Rao, S.; and Daumé III, H. 2019. Answer-based Adversarial Training for Generating Clarification Questions. 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)*, 143–155.

Shen, T.; Ott, M.; Auli, M.; and Ranzato, M. 2019. Mixture models for diverse machine translation: Tricks of the trade. *arXiv preprint arXiv:1902.07816* .

Shi, J.; and Malik, J. 2000. Normalized cuts and image segmentation. Technical report.

Song, L.; Wang, Z.; Hamza, W.; Zhang, Y.; and Gildea, D. 2018. Leveraging context information for natural question generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 569–574.

Stoyanchev, S.; Liu, A.; and Hirschberg, J. 2014. Towards natural clarification questions in dialogue systems. In *AISB symposium on questions, discourse and dialogue*, volume 20.

Subramanian, S.; Wang, T.; Yuan, X.; Zhang, S.; Bengio, Y.; and Trischler, A. 2018. Neural Models for Key Phrase Extraction and Question Generation. *ACL 2018* 78.

Sun, X.; Liu, J.; Lyu, Y.; He, W.; Ma, Y.; and Wang, S. 2018. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3930–3939.

Vijayakumar, A. K.; Cogswell, M.; Selvaraju, R. R.; Sun, Q.; Lee, S.; Crandall, D.; and Batra, D. 2018. Diverse beam search for improved description of complex scenes. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Wang, Y.; Liu, C.; Huang, M.; and Nie, L. 2018. Learning to Ask Questions in Open-domain Conversational Systems with Typed Decoders. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2193–2203.

Zamani, H.; Dumais, S. T.; Craswell, N.; Bennett, P.; and Lueck, G. 2020. Generating Clarifying Questions for Information Retrieval.