Atomic Self-Consistency for Better Long Form Generations

Anonymous ACL submission

Abstract

Recent work has aimed to improve LLM generations by filtering out hallucinations, thereby improving the precision of the information in responses. Correctness of a long-form response, however, also depends on the recall of multiple pieces of information relevant to the question. In this paper, we introduce Atomic Self-Consistency (ASC), a technique for improving the recall of relevant information in an LLM response. ASC follows recent work, Universal Self-Consistency (USC) in using multiple stochastic samples from an LLM to improve the long-form response. Unlike USC which only focuses on selecting the best single generation, ASC picks authentic subparts from the samples and merges them into a superior composite answer. Through extensive experiments and ablations, we show that merging relevant subparts of multiple samples performs significantly better than picking a single sample. ASC demonstrates significant gains over USC on multiple factoids and open-ended QA datasets - ASQA, QAMPARI, QUEST, ELI5 with ChatGPT and Llama3. Our analysis also reveals untapped potential for enhancing long-form generations using approach of merging multiple samples.

1 Introduction

Large language models (LLM) with their ability to perform mathematical reasoning (Wei et al., 2022), planning (Ahn et al., 2022), and generating human-like text (Bubeck et al., 2023) have become an integral component of many AI systems. Long-form question answering (LFQA) is an important benchmark task whose performance reflects the reliability of these AI systems at providing comprehensive and accurate responses to user queries.

In LFQA, each response comprises multiple pieces of information, described in the literature as atomic facts (Min et al., 2023), that collectively contribute to the overall correctness of the answer. Despite various improvements, LLMs are still very prone to producing hallucinatory content such as incorrect atomic facts, especially when the responses are longer (Ren et al., 2023). Recent works on mitigating hallucinations have primarily involved the removal of inaccurate atomic facts from the generated content. While these methods produce responses with high precision over atomic facts, the correctness of the response also depends on the inclusion of all information relevant to the question, i.e., the recall of atomic facts relevant to the question. E.g. In Fig. 1, A_1 is a precise response. A_2 is a more complete high recall response to Q.

> Q: What are the main causes of climate change?
>
> A1: The primary causes of climate change include human activities like burning fossil fuels, deforestation, and industrial processes.
>
> A2: The primary causes of climate change include human activities like burning fossil fuels, deforestation, and industrial processes. Natural factors such as volcanic eruptions and changes in solar radiation also contribute.

Figure 1: A_1: A precise answer. A_2: An answer with higher recall of atomic facts relevant to the question Q.

On the other hand, in QA with closed-form answers (such as a math problem with a numeric answer), remarkable improvements in response quality have been achieved by stochastically sampling multiple model responses and then using consistency criteria to select one as the final answer (Wang et al., 2022). Recently, similar efforts were extended to long-form generation. Universal Self Consistency (USC) (Chen et al., 2023), is one such example which uses LLMs to determine consistency between model responses. The output of USC is the single most consistent generation among multiple candidate samples from the model.

However, picking a single final answer among the candidate generations might miss out on relevant atomic facts from other candidates and not optimize the recall of information. Further, it is still prone to some atomic hallucinations within the final selected candidate. To overcome these challenges, we propose a simple approach called

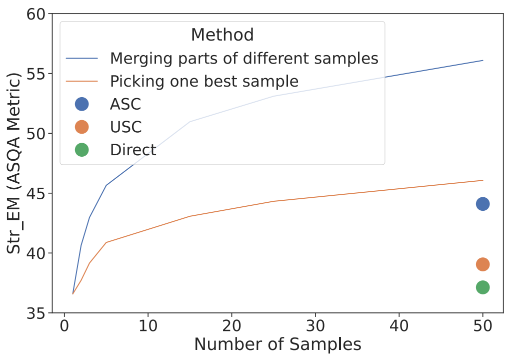

Figure 2: Best possible recall (oracle performance) with increasing number of samples on ASQA(ChatGPT). Merging subparts from multiple samples has a much higher ceiling. ASC beats USC, Direct; almost matches the ceiling performance of picking one best sample.

Atomic Self-Consistency (ASC), which combines authentic atomic facts from multiple candidate responses to generate a superior composite response. To motivate the readers on the potential benefits of this approach, Fig. 2 shows the oracle performance (best possible performance) of picking one single generation vs merging subparts of multiple generations on the ASQA dataset (details in §4.7). Merging answers from multiple samples have significant performance potential over picking a single answer. USC would not be able to capture this potential. Fig. 2 also shows the performance of ASC and other baselines. ASC matches the ceiling performance possible of picking the best sample.

Key Contributions 1. We introduce a simple and efficient method, ASC, that: (i) clusters atomic parts of multiple candidate generations, (ii) uses a consistency-based criterion to pick the best clusters relevant to the question, and (iii) finally combines them all into a single superior answer. ASC operates in black box mode and does not require access to LLM weights or logits. 2. We systematically establish the superiority of combining multiple generations over picking a single generation. 3. We extensively evaluate and show significant performance improvements of ASC over USC and other hallucination reduction baselines on four diverse long-form QA tasks—ASQA, QAMPARI, QUEST and ELI5. We justify the benefits of ASC through rigorous ablations. 4. We show strong empirical evidence for minimizing the number of stochastic samples required by ASC. 5. Finally, our analysis reveals untapped potential for enhancing LLMs by merging multiple generations into a superior composite output. This insight underscores a promising avenue for advancing LLM performance further.

2 Related Work

We discuss key methodologies that are used to identify hallucinations and alleviate hallucinations. We further talk about the importance of *consistency* as a measure for correctness and finally talk about how this is used to improve LLM response.

Detecting and alleviating hallucination: FactScore (Min et al., 2023) proposed a mechanism to identify atomic facts using an LLM and then individually check each fact's correctness using a retrieval-based solution. CoVe (Dhuliawala et al., 2023) used the LLM to generate multiple verification questions over a candidate generation and prompted an LLM to answer them. Answers to these verifying questions were used to draft a high-precision response. Agrawal et al. (2023) used indirect prompts to verify individual units in list-style answers.

Consistency as a measure for correctness: Consistency between model responses resulted in performance leaps in mathword problems, code generation, etc (Xiong et al., 2022; Shi et al., 2022). SelfCheckGPT (Manakul et al., 2023) is another work very relevant to ASC which detects hallucination in model responses. It verifies the correctness of individual sentences in a generation by measuring their agreement with multiple other stochastic samples by the LLM. It showed the benefit of consistency-based measures to identify sentence-level hallucination, within long-form generations. HaLo (Elaraby et al., 2023) also used consistency-based measures to identify sentence-level hallucinations in a generated text. It also explored techniques like knowledge injection and distillation.

Using stochastic samples to improve LLM response: USC (Chen et al., 2023) uses the consistency-based measure to pick the most consistent individual response among stochastically generated sample responses. It feeds in all responses to the LLM and asks to output the most consistent response. Similar to USC, Ren et al. (2023) pick one best answer from a list of candidates by a hybrid mechanism which contains a score from the samples posed as a multiple choice question and a score based on self evaluation from the same LLM.

Despite remarkable improvements, these methods are confined to either filtering atomic facts from a response or picking a single final answer from multiple samples. Our research on the other hand, focuses on combining subparts of multiple stochastic samples to produce higher quality generations.

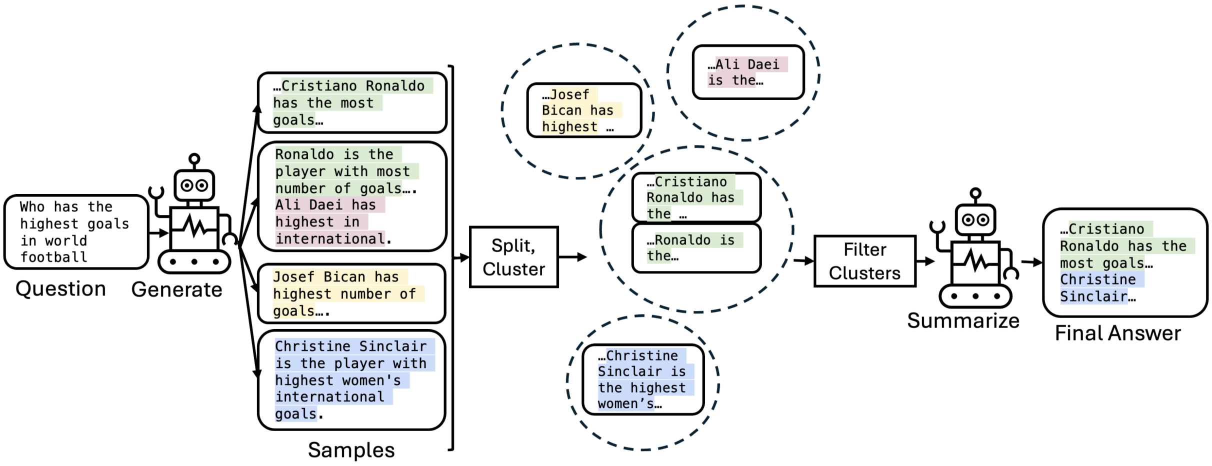

Figure 3: Overall Pipeline proposed. Generated samples are split into smaller parts and clustered. Clusters are then filtered by a consistency based criterion (higher strength clusters are selected while lower strength clusters are removed). Selected cluster representatives are then summarized by an LLM to generate a final answer.

3 Methodology

Given a question q, our task is to use an LLM $\mathcal{L}$ to produce an answer which answers the question both accurately (i.e., with high precision) and comprehensively (i.e., with high recall). We describe the specific metrics used to measure this for each dataset in §4. Let $a_1, a_2, .., a_m$ be m independent samples directly generated by $\mathcal{L}$ when query q is fed to it in a prompt P. This work merges consistent subparts from the multiple samples to produce a final answer a_{ASC} in a four-step process. 1. Split: Split each generation into its constituent atomic facts. 2. Cluster: Grouping atomic facts for efficiency 3. Filter: Selecting the most consistent clusters. 4. Summarize: Combine the selected cluster representatives to generate a final answer.

3.1 Splitting Generations for Atomic Facts

Since atomic facts can be verified for their truth, the first step in our approach is to break down candidate generations into atomic facts. A candidate generation to a question might comprise multiple sentences and multiple atomic facts within each sentence. Min et al. (2023) used an InstructGPT model to break down a long-form generation into its atomic facts. While we believe that use of such neural models can produce better quality atomic facts, it is extremely expensive in our setting as this needs to be performed for m different generations per question. Hence, following Arslan et al. (2020); Manakul et al. (2023); Liu et al. (2023), we used individual sentences within a candidate generation as its atomic facts. We used sentence tokenization models (Bird et al., 2009) to split the generation into its constituent sentences (atomic facts).

3.2 Clustering Atomic Facts

Each of the m generations typically contains multiple individual sentences/atomic facts, say k on average. Each of these units needs to be verified for relevance and truthfulness before inclusion in the final answer. However, evaluating each atomic unit either by external sources like retrieval or by prompting $\mathcal{L}$ would require $m * k$ steps and prove very expensive. Note that multiple of these atomic units share content because they were generated addressing the same question. We hence perform a round of clustering to collect atomic facts conveying the same meaning. Despite being cubic in computational complexity, we found agglomerative clustering (Pedregosa et al., 2011) of sentence embeddings obtained from SimCSE (Gao et al., 2021) has less overhead and is much faster compared to using retrieval/LLM calls to filter them.

Given the clusters, verifying only the representatives of each cluster would contribute to substantial savings in compute and time. We chose the longest sentence in each cluster as its representative.

3.3 Filtering Clusters

The objective now is to identify and eliminate inaccurate atomic facts, ensuring that only valid and reliable atomic facts are retained for the final answer generation in the subsequent steps. Multiple methods have been used in the literature to verify facts, e.g., retrieval-based verification Min et al. (2023) and self-evaluation Manakul et al. (2023). These methods are still expensive to perform even if it's just cluster representatives that we evaluate. Consistency between model responses when used as a metric resulted in significant gains in reasoning. In our case, a measure of consistency is readily available in the form of the clusters' strengths. Hence, we use the individual cluster strengths to pick the most consistent of them. Specifically, all clusters having count below a fixed threshold Θ (tuned on a validation set) are filtered and the consistent clusters' representatives are selected $\{c_1, c_2, c_3, ..c_z\}$. We also experimented with self evaluation in our preliminary analysis but found consistency-based measures worked better. We also compare with retrieval-based method to filter clusters in the §4.

3.4 Summarizing Selected Clusters

The representatives $\{c_1, c_2, c_3, ...c_z\}$ are fed into the same LLM $\mathcal{L}$ with a summarizing prompt $P_{combine}$(mentioned in §A) to produce the final answer $a_{\texttt{ASC}}$. Note that the number of representatives $z < m$ (most times) and hence is much easier to process by $\mathcal{L}$ compared to USC (feeds m inputs). This call to the LLM not only summarizes the selected cluster representatives but also filters any slack/filler sentences that were selected by the consistency metric. Overall pipeline is shown in Fig. 3.

Adapting ASC to list style datasets: We also extend the ASC methodology to list style datasets. Here, for a given question, the answer produced is typically a list of entities. Following Agrawal et al. (2023), we directly used each item in the list as an atomic fact. We used surface-form based clustering where two atomic units are placed in the same cluster if their normalized edit distance is below a threshold. Θ threshold (tuned on validation set) based filtering is applied to select consistent clusters. The first item in each cluster is considered its representative. The final answer is just the list of selected cluster representatives $[c_1, c_2, c_3, ...c_z]$.

4 Experiments

ASQA (Stelmakh et al., 2022): A long-form factoid dataset comprising ambiguous questions. Following Gao et al. (2023), performance on this dataset is evaluated by 1. 'Str_EM': checking if reference short answers have an exact match in the LLM generated answer, 2. 'QA-F1': Does an external QA model identify these short answers from reference disambiguating questions. Str_EM is very closely related to the recall of atomic facts relevant to the question. Additionally, we also present the 'Mauve' score which compares the fluency and style of the model generated text to the reference answer. **QAMPARI** (Rubin et al., 2022): QAMPARI is a list-style factoid QA dataset constructed from Wikipedia knowledge graphs and tables with the questions paraphrased by humans. **QUEST** (Malaviya et al., 2023): QUEST is another list-style dataset constructed using Wiki category lists. 'Precision', 'Recall', 'F1', 'Recall-5(100 if atleast 5 correct entities are present)' are used to evaluate QAMPARI and QUEST. **ELI5** (Fan et al., 2019): A long-form QA dataset containing how/why/what questions from Reddit. We use 'Claims_Nli' of golden subclaims following Gao et al. (2023). Details on datasets and metrics in §A.2.1.

4.1 Methods

We compare the following methods. The first two are *strong hallucination reduction* methods. The later ones include *stochastic sampling* methods.

ACF (adapted from **SelfCheckGPT**, (Manakul et al., 2023)): Abbreviated from Atomic Consistency-based Filter. Only one generation (very first seed) out of the m is used. We use cluster strengths from the m generations to filter out facts from the one generation. Selected facts are combined as mentioned in §3.4.

FCF (adapted from **FActScore**, (Min et al., 2023)): Abbreviated from Factual Correctness based Filter. Uses FactScore (Min et al., 2023) filter to throw out facts from the one generation similar to ACF. Leftover facts are combined as mentioned in §3.4.

Direct: Direct generation from the LLM. Results are averaged over five different seeds.

USC-LLM: This is the original formulation which used a list of samples as input to the LLM and found the most consistent response. We reduced the input list when it did not fit the context window. We observed that this method picks the first response a majority of the time possibly because of LLM's longer list problems (Qin et al., 2023).

USC: Consistency based method proposed in Chen et al. (2023). To overcome LLM list problems in USC-LLM, we use the same clustering pipeline as mentioned for ASC. Each of the m generations is given a score equal to the sum of the strengths of all clusters it contributes to. The highest scoring generation is selected as the final answer.

ASC: The method that we propose in this work. Splits multiple samples into smaller parts, and clusters them. The best clusters (picked by consistency score) are summarised using an LLM.

ASC-F: Similar to ASC but uses FActScore to pick the relevant clusters instead of the cluster strength

		ASQA					ELI5			
		#Clusters	length	Mauve	Str_EM	QA-F1	#Clus.	length	Mauve	Claims_Nli
ChatGPT	Direct	-	56.29	44.64	37.13	29.33	-	104.35	24.57	18.66
	ACF		42.99	53.66	36.16	28.98		84.11	20.73	18.2
	FCF		45	52.68	36.84	29.64		94.75	27.97	18.7
	USC-LLM		56.72	44.88	37.91	29.71		104.13	21.11	18.76
	USC		64.52	40.19	39.05	30.88		97.36	24.09	17.4
	ASC-F (Ours)	30.74	106.7	41.25	**44.96**	31.91	56.83	172.66	22.68	**22.16**
	ASC (Ours)	15.7	101.17	47.01	44.1	**32.22**	16.68	163.58	21.29	21.43
Llama3-70b	Direct	-	40.68	62.31	35.46	28.67	-	95.76	26.77	18.09
	ACF		34.67	54.01	35.65	29.48		83.1	29	19.73
	FCF		34.73	59.37	35.71	29.41		82.21	25.32	19.53
	USC		55.26	47.33	38.29	30.11		128.55	21.14	21.97
	ASC-F	11.94	71.41	69.25	**41.89**	**31.19**	51.86	167.11	22.91	23.63
	ASC	7.57	66.88	71.16	40.97	30.61	28.17	164.91	21.06	**23.73**

Table 1: ASQA, ELI5 results. ASC does the very well on QA-F1 and demonstrates strong Str_EM. ASC-F picks a large number of clusters and also does well. ASC also demonstrates strong Mauve. ASC, ASC-F achieve best Claims_Nli score on ELI5. Results justify that merging of samples is better than picking one sample.

		QAMPARI						QUEST					
	Method	#Pred	Prec	Rec	Rec-5	F1	F1-5	#Pred	Prec	Rec	Rec-5	F1	F1-5
ChatGPT	Direct	5.2	21.35	13.82	23.47	15.35	21.83	5.56	12.05	6.76	12.91	7.45	11.6
	ACF	3.61	24.16	12.5	21.96	15.04	22.18	3.07	14.71	5.65	10.67	7.06	11.53
	FCF	4.41	22.59	13.29	23.16	15.33	22.16	3.61	13.55	5.91	11.03	7.01	11.27
	USC-LLM	4.95	20.88	13.39	22.91	14.94	21.33	5.10	11.86	6.18	11.92	7.08	11.16
	USC	8.97	20.7	19.21	31.28	18.07	24.2	7.83	11.98	8.43	15.19	8.23	12.21
	ASC-F	40.83	13.42	29.81	45.04	15.7	18.82	39.9	7.94	17.31	30.73	8.47	10.84
	ASC	7.09	22.98	20.5	33.04	**19.46**	**26.21**	8.44	12.47	10.41	19.15	**9.75**	**14.09**
Llama3-70b	Direct	4.65	25.84	14.57	25.29	16.85	23.94	4.62	13.16	5.95	11.28	7.07	11.21
	ACF	3.43	25.04	11.99	21.12	14.86	21.55	3	13.88	4.91	9.3	6.34	10.33
	FCF	4	26.71	13.79	23.82	16.47	23.5	3.08	15.5	5.32	10.07	6.8	11.15
	USC	7.23	23.93	18.35	30.16	18.79	25.07	7.65	12.97	8.01	14.65	8.27	12.37
	ASC-F	24.65	17.8	26.72	41.72	18.28	22.24	30.82	8.97	13.82	25.77	8.64	11.37
	ASC	9.61	22.94	22.09	35.86	**20.16**	**26.02**	11.75	11.04	10.24	19.21	**8.94**	**12.6**

Table 2: ASC outperforms Direct, USC and ASC-F. ASC-F picks a large number of clusters and does worse on P, F1, F1-5. Results justify that consistency-based cluster selection does better than retrieval-based cluster selection.

in §3.3. We use 5 retrieved passages and InstructGPT from FactScore to verify the correctness of each cluster. ASC-F doesn't use any consistency measures despite sharing the name.

4.2 Model Details and Setup

We use $m = 50$ for generations. The same set of generations are used by ASC, ASC-F, USC. Direct is the average of five seeds among the 50. ACF, FCF use only one seed for the answer. ACF uses all 50 seeds but for building a consistency-based selection mechanism. Sentence embeddings for clustering were generated using robert-large SimCSE (Gao et al., 2021), agglomerative clustering ($d = 0.15$) is used to perform clustering in ASQA, ELI5. More details on models/hyperparams/Θ used in §A.1

4.3 Results

Table 1 demonstrates results for ASQA. ASC, ASC-F do much better than USC in Str_EM (exact match) and QA-F1 (QA model performance). This shows that merging parts of multiple answers performs better than picking a single answer. ACF and FCF are more picky in selecting what facts can show up in the final answer and hence have worse Str_EM, QA-F1 even compared to Direct in some cases. The short answers in ACF, FCF help achieve a better Mauve scores. ASC beats Direct, USC, and ASC-F in achieving better Mauve score. Table 1 also shows the eval set results of ELI5. ASC, ASC-F perform better than all other models yet again demonstrating the strength of merging multiple model responses. Mauve is lower in this case because the reference answers are from Reddit subposts and the style didn't match the long answers generated by ASC. As will be shown in ablations, ASC offers easy control over Mauve by changing Θ. Fig. 7 shows that Θ can be adjusted to improve ASC's Mauve score over others while retaining Claims_Nli.

Ablation	Method	ASQA					QAMPARI					
		#Clusters	length	Mauve	Str_EM	QA-F1	#Pred	Prec	Rec	Rec-5	F1	F1-5
1	`ASC`	15.7	101.17	47.01	**44.1**	**32.22**	7.09	22.98	20.5	33.04	**19.46**	**26.21**
	`Random Clusters`	15.7	85.31	49.97	<u>42.62</u>	<u>31.75</u>	7.09	11.86	10.08	18.62	9.77	14.05
	`Random Sentences`	15.7	99.45	42.08	41.5	29.36	7.09	22.19	13.8	24.42	15.39	22.1
2	`USC`	-	64.52	40.19	39.05	30.88	8.97	20.7	19.21	31.28	<u>18.07</u>	<u>24.2</u>
	`High Token/#Pred`		82.93	40.59	37.8	28.79	10.48	17.19	18.3	29.28	16.07	21.01

Table 3: Ablation 1: ASC performs better than randomly picking clusters and randomly picking sentences on ASQA, QAMPARI. Ablation 2: Larger length or higher #Predictions in response is not critical for better performance.

Table 2 shows test set results for QAMPARI. `ASC` performs the best. `ASC-F` similar to the previous case selects a large number of clusters. It does well on recall but significantly drops on precision leading to a worse overall performance. This also shows that longer answers are not always helpful. Since this is a list-style dataset, we also show #Predictions (size of the list) which is somewhat equivalent to #Clusters in the previous case. Note that `ASC` beats `USC` despite having lower #Predictions with ChatGPT. As expected, `ACF`, `FCF` have much lower #Predictions and have higher precision compared to others. `ASC` relies on consistency to predict a much higher #Predictions while also matching/improving precision. This strongly justifies the strength of consistency-based cluster selection. Specifically, it achieves the two goals we had set out in this exploration. It removes incorrect atomic facts from a sample (increase in precision compared to `USC`) and adds correct atomic facts from other samples (increase in recall compared to `USC`). Table 2 also shows test set results for QUEST. The trends are very similar to QAMPARI with `ASC` performing the best. Summarizing all the observations from above, we conclude that

1. Merging LLM samples is better than selecting one single sample.
2. Atomic-Consistency is a strong measure to select clusters and improve correctness.

4.4 Ablations

4.4.1 Ablation 1: Dissecting ASC

To effectively understand the contribution of different components of ASC, we analyze the effect of each subcomponent. Table 3 shows results with ASQA. ASC first clusters individual sentences from all 50 generations and merges sentences with high cluster strength using an LLM. The `Random Clusters` method follows the same clustering as `ASC` but randomly picks clusters before merging them using an LLM. `Random Sentences` doesn't perform any clustering and randomly picks the sentences from all generations and summarizes them. In both runs, we pick the exact same number of sentences that were picked by `ASC`. `Random Clusters` drops both Str_Em and QA-F1 but still does better than `Direct`, `USC` in Table 3. This shows the strength of including diverse sentences from multiple samples into the answer generation. In the case of ASQA and ELI5, there is less hallucination compared to QAMPARI and QUEST. Hence, randomly picking clusters does fairly well better than most other baselines. `Random Sentences` further drops in metrics while still maintaining a high Str_EM.

Table 3 also does the same analysis for QAMPARI. Here again, we run the two random baselines. Similar to the previous case, `ASC` does the best.

4.4.2 Ablation 2: Longer length answers

From Table 2, 1, `ASC` and `ASC-F` often have higher length answers and higher #predictions. One might deduce that longer generations tend to give better results on the datasets tested. Hence, we perform an additional experiment which picks the longest length sample (among the 50 samples) for ASQA and pick the sample with the highest #Predictions (among the 50 samples) for QAMPARI as the final answer. Results are shown in Table 3. Despite having a larger length or higher #predictions, `USC` with a lower length and lower #predictions perform much better. This shows length is not the most important factor for improved performance.

4.4.3 Ablation 3: ASC is simple to control with Θ (Sensitivity Analysis)

Θ is a parameter which critically affects the performance of `ASC`. For Table 2, we used a value of Θ that performed the best on the validation set. Different number of selected clusters result in differences in various performance metrics. For example, `ASC-F` which selects a large number of clusters is more suited to high recall scenarios where precision is less important. Hence, to better understand this effect, we experiment with different values of Θ in this subsection. Fig. 4 shows the effect of varying Θ on ASQA. A lower Θ resulted in selecting a large number of clusters and resulted in improving QA-F1. This also increased the length of the final response. Reducing Θ on the other hand improved the Mauve fluency score as the shorter final answer matched more with the reference answer. Hence, one might easily adjust Θ to obtain an answer aligned with their preference (Mauve or QA-F1). From the Fig. 4, ASC can outperform all other methods in Mauve(can achieve >65) while still retaining a QA-F1 (>31). The best of other methods was Mauve (53.66) and QA-F1 (30.99). A similar result was seen in ELI5 where increasing Θ achieved the highest Mauve §7.

Fig. 6 in §A shows the effect of varying Θ on QAMPARI. The relationship here is more linear. Increasing Θ results in fewer clusters with high strength (high precision). Reducing Θ results in higher recall. ASC-F's criterion enabled it to select a larger number of clusters resulting in higher recall. Here again, one can easily change Θ to obtain an answer with preferred qualities.

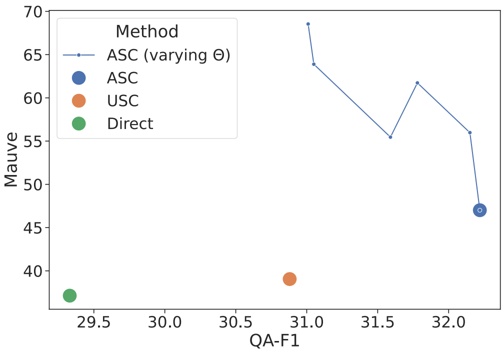

Figure 4: ASQA. Increasing Θ improves QA-F1, reduces Mauve. Adjusting Θ produces a preferred answer.

4.5 Analysis: Can ASC work with fewer number of generations? Use Entropy

Cost of generation using an LLM linearly scales with the number of samples. ASC used a large number of samples, $m = 50$ in our earlier experiments. It might not always be feasible to generate this large number of samples due to time and budget constraints. In this subsection, we investigated if we could generate fewer samples and yet capture the gains provided by ASC. While we focused on QAMPARI for this analysis, we found similar trends with other datasets as well. We first looked at how the entropy of the clusters (considering each cluster to have a probability proportional to its strength) changes with increasing number of generations. In the beginning ($m = 1$), all clusters have one member and equal probability. Hence, the entropy is lower. As and when more samples get

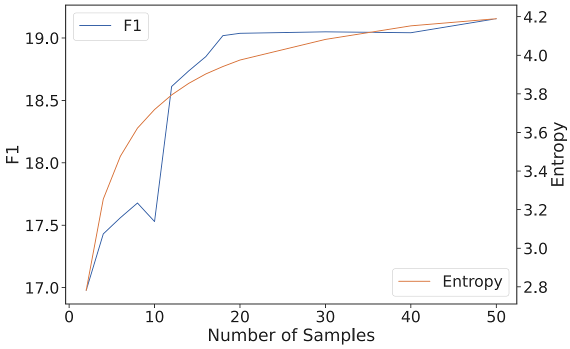

Figure 5: QAMPARI. Performance starts to stagnate when clusters' entropy stagnates.

added, some clusters accumulate higher strength and some others remain low strength. Hence, the entropy increases due to unequal probabilities of clusters. We empirically found that entropy starts to stagnate with higher values of m. To measure the performance of m samples, we scaled the optimal Θ we found in Table. 2 by $\frac{m}{50}$. We found that performance follows a similar trend increasing quickly at the beginning while slowly stagnating. Performance and Entropy curves values with m are shown in Fig. 5. Interestingly, performance starts to stagnate right around when entropy starts to stagnate. Entropy stagnation can thus be used as an indication to stop generating more samples from the LLM and fix m.

4.6 Analysis: Clustering

Clustering being a core component of ASC, we perform an extensive quantitative and qualitative analysis over it. Firstly, we try multiple embedding and clustering methods results of which are shown in Table 8 §A.4. Note that the results are consistent across different choices fo embedding models and clustering methods thus justifying our earlier choice of embedding models/clustering methods.

Quantitative Analysis: Further, for each cluster in all examples, we pass all constituent atomic facts (sentences in our case) to GPT4 and zero-shot ask " You are given a list of sentences. What percentage of them convey similar meaning?". We parse a number(%) from this and average it over all clusters of all examples and present it as Purity in Table 4. Across different embedding and clustering methods, we observe high purity of clusters.

Qualitative Analysis: For further qualitative analysis, we demonstrate clustered example question from ASQA in Table 9 of §A.4. As can be seen each cluster contains sentences that convey similar meaning. Eventhough the meaning is similar, they

Embedding	Clustering	Purity (%)
$\text{Simcse}_{\text{Roberta\_Large}}$	Agglomerative	97.32
$\text{Simcse}_{\text{Roberta\_Large}}$	KMeans	96.48
$\text{GTR}_{\text{t5-xxl}}$	Agglomerative	96.23
$\text{GTR}_{\text{t5-xxl}}$	KMeans	95.04

Table 4: Purity of Different Embeddings and Clustering Methods. Clustering parameters set to approximately result in same number of clusters i.e. d=0.15 and K=39.

might contain some slightly different facts (E.g. exact number of goals scores in Cluster 3). Sometimes, we found that sentences conveying similar meaning formed more than one cluster - For Ex: Cluster 3 and Cluster 5. We left it for the LLM summarization step to filter out such repetitions. We pick one representative from each cluster and sent them to the LLM for summarization.

4.7 Analysis: Room for improvement

		ASQA		*QAMPARI*	
Method	**#Gen**	**Str_EM**	**QA-F1**	**Rec**	**Rec-5**
Oracle	1	36.32	22.88	13.94	24.24
	2	40.64	28.05	18.15	30.46
	5	45.65	34.03	24.53	39.02
	15	50.97	39.28	32.29	48.78
	25	53.1	41.29	35.86	52.76
	50	56.09	45.2	40.06	56.90
ASC	50	44.1	32.22	20.50	33.04

Table 5: Oracle results reveal sizable scope for improvement using our approach of merging multiple responses.

To better understand the gains of ASC, we look at the best possible performance offered by our mechanism of merging multiple sample generations. We use the same 50 generations that were used to produce the ASC results. Table 5 shows the best possible performance (Oracle) with the number of generations. Exact procedure for obtaining the oracle numbers is described in §A.5.

The experiment presents interesting observations. 1. Using just five generations significantly increases the oracle performance. 2. Oracle's performance stagnates at a higher number of generations. Our observations on ASC performance stagnating after 20 generations are in line with these results. 3. ASC only captures 20.50 of the 40.06 possible recall on QAMPARI and 44.1 of the 56.09 possible Str_EM on ASQA. Thus while ASC captures a fair share of the performance gain offered by merging multiple generations, a sizable portion of performance gain still remains untapped. Future work aims at capturing this potential gain by using stronger verification methods involving a combination of ASC with ASC-F and methods in §2.

5 Discussion

5.1 Hallucination Reduction Methods vs ASC

We compared with adaptations of two strong hallucination reduction methods in FCF, ACF - FactScore (Min et al., 2023), Self CheckGPT (Manakul et al., 2023) respectively. Hallucination reduction methods like Dhuliawala et al. (2023), Ren et al. (2023) operate similarly in terms of removing any hallucinatory facts and retaining correct facts in the generated answer (improved precision). These methods lag in recall as shown in table. 2. In contrast, ASC additionally captures authentic content from other generations which was not included in the original answer.

5.2 Stochastic Sampling Methods vs ASC

As shown in Fig. 2, merging best subparts of multiple generations has significantly higher scope over picking the single best generation. Hence, ASC does better than otehr stochastic sampling methods like Ren et al. (2023), USC.

Runtime Analysis: The exact same number of LLM calls are required by ASC (50 generation + 1 summarization) and USC (50 generation + 1 consistent answer picking). While ASC additionally requires extra compute to perform clustering, this can be done using smaller language models on sentences and is less costly. In contrast, Ren et al. (2023) uses (50 + multiple) LLM calls to select the best answer and hence is more expensive.

6 Conclusion

In this work, we propose ASC, a simple way of merging subparts of multiple answer samples produced by an LLM. Through extensive experiments and ablations, we show the 1. Benefits of merging subparts of multiple answers over picking one single answer. 2. Strength of *consistency* as a measure for improving correctness. ASC significantly outperforms USC, a strong baseline for generating long-form answers. We show empirical evidence for minimizing the number of samples required by ASC. Finally, our analysis also reveals untapped potential for enhancing long-form generations using our approach of merging multiple responses.

7 Limitations

Smaller language models not tried Some of the datasets used in our work are very challenging and not suitable for smaller language models. To effectively prove the strength of our approach, we stuck to ChatGPT, Llama-70b.

Multiple samples still need to be generated A general limitation of most stochastic sample based methods. These methods rely on generating multiple samples and picking the final answer among them. However, this might be slightly expensive. Speculative Decoding (Li et al., 2024) has recently made great strides in reducing the amount of compute required to sample from Large Language Models. Speculative decoding can be used to significantly reduce the compute required by ASC and USC.

Broader Impact and Discussion of Ethics: While our model is not tied to any specific applications, it could be used in sensitive contexts such as health-care, etc. Any work using our method is requested to undertake extensive quality-assurance and robustness testing before applying in their setting. To the best of our knowledge, the datasets used in our work do not contain any sensitive information.

License: Refer to the licenses of individual training datasets used Stelmakh et al. (2022), (Rubin et al., 2022), (Malaviya et al., 2023), (Fan et al., 2019) and LLM models used Touvron et al. (2023), (Achiam et al., 2023).

Replicability:
Code and Datasets used will be made publically available.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ayush Agrawal, Lester Mackey, and Adam Tauman Kalai. 2023. Do language models know when they're hallucinating references? *arXiv preprint arXiv:2305.18248*.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Fatma Arslan, Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2020. A benchmark dataset of check-worthy factual claims. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 821–829.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models.

Mohamed Elaraby, Mengyin Lu, Jacob Dunn, Xueying Zhang, Yu Wang, and Shizhu Liu. 2023. Halo: Estimation and reduction of hallucinations in open-source weak large language models. *arXiv preprint arXiv:2308.11764*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*.

Nelson F Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. *arXiv preprint arXiv:2304.09848*.

Chaitanya Malaviya, Peter Shaw, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2023. Quest: A retrieval dataset of entity-seeking queries with implicit set operations. *arXiv preprint arXiv:2305.11694*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

Jie Ren, Yao Zhao, Tu Vu, Peter J Liu, and Balaji Lakshminarayanan. 2023. Self-evaluation improves selective generation in large language models. *arXiv preprint arXiv:2312.09300*.

Samuel Joseph Amouyal Ohad Rubin, Ori Yoran, Tomer Wolfson, Jonathan Herzig, and Jonathan Berant. 2022. Qampari:: An open-domain question answering benchmark for questions with many answers from multiple paragraphs. *arXiv preprint arXiv:2205.12665*.

Freda Shi, Daniel Fried, Marjan Ghazvininejad, Luke Zettlemoyer, and Sida I Wang. 2022. Natural language to code translation with execution. *arXiv preprint arXiv:2204.11454*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. *arXiv preprint arXiv:2204.06092*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jing Xiong, Zhongwei Wan, Xiping Hu, Min Yang, and Chengming Li. 2022. Self-consistent reasoning for solving math word problems. *arXiv preprint arXiv:2210.15373*.

A Appendix

A.1 Implementation Details

For ASC-F, FCF, retrieval, query and document embeddings were generated using GTR-T5-XXL (Ni et al., 2021) and Wikipedia following (Gao et al., 2023). In ASC-F, FCF, each cluster/fact used 5 retrieved passages for verification. InstructLlama model from Min et al. (2023) was used to verify facts in ASC-F,FCF. A fact was called true if at least 1 of the 5 passages supported it. Experiments on all four datasets were performed with both ChatGPT, Llama-3(70b), Llama-2(70b) (Touvron et al., 2023). In all the experiments, the same LLM is used to perform both generation and summarization. Generation and summarization prompts along with other details are presented in §A.1.

ASC uses hyperparam Θ tuned over the development set to maximise F1-5 for QAMPARI and QUEST. For ASQA, ELI5, Θ was chosen such that the number of selected clusters comfortably fit the context window of ChatGPT, Llama3. ACF used the same threshold as ASC in filtering atomic facts. FCF, USC did not require tuning any hyperparameters. Clustering parameters were same for all models that used it.

A.2 Runtime Details

We followed (Gao et al., 2023) to generate 50 responses from ChatGPT and Llama. We used four 48gb A6000 gpus for all experiments. Generating responses using ChatGPT was much faster and only took 3hrs per dataset. Generating the responses with Llama2/3 was much more challenging and took 24 hrs per dataset.

As ASC only contains simple clustering steps, it runs fairly fast with an average of 3hrs per dataset with ChatGPT. ASC with Llama includes the final summarizarion step which took 15 hrs on average over datasets.

A.2.1 Tasks

ASC did not use the training set of any of these datasets.

ASQA (Stelmakh et al., 2022): ASQA is a long-form factoid dataset comprising ambiguous questions. The ambiguous nature of the questions requires answers to comprise diverse facts from multiple documents. The dataset provides individual reference disambiguating short answers for each question and also a reference long answer combining all short answers. Evaluation was done on the eval set (948 examples). Following Gao et al. (2023), performance on this dataset is evaluated by 1. ‘Str_EM’: checking if reference short answers have an exact match in the LLM generated answer, 2. ‘QA-F1’: Does an external QA model identify these short answers from reference disambiguating questions. Str_Em is very closely related to the recall of atomic facts relevant to the question. Additionally, we also present the ‘Mauve’ score which compares the fluency and style of the model generated text to the reference answer.

QAMPARI (Rubin et al., 2022): QAMPARI is a list-style factoid QA dataset constructed from Wikipedia knowledge graphs and tables with the questions paraphrased by humans. Performance over this dataset is evaluated by ‘Precision’, ‘Recall’ and ‘F1’ between the generated answer list and reference answer list. As the reference lists are often huge, another measure ‘Recall-5’ scores the answer 100 if at least 5 correct entities are present. Evaluation was done on the test set with 1000 examples.

QUEST (Malaviya et al., 2023): QUEST is another list-style dataset constructed using Wiki category lists. This is a much more challenging dataset compared to QAMPARI. Following Dhuliawala et al. (2023), we transform each category name into a question by prepending "*Name Some*". For Eg. *"Name Some Mary Stewart novels"*. Performance over this dataset is evaluated by Precision, Recall, F1 and Recall-5. Evaluation was done on the test set with 1727 examples.

ELI5 (Fan et al., 2019): This is a long-form QA dataset containing how/why/what questions from Reddit. Gao et al. (2023) had generated three sub-claims from each golden answer and showed that an answer’s entailment score over these sub-claims provides a more accurate measure of its correctness. We use this same ‘Claim-Recall’ to measure the correctness of a generated answer in this work. Similar to Str_EM in ASQA, this again is very related to the recall of atomic facts relevant for the question. We use the same randomly sampled 1000 questions from the eval set as from Gao et al. (2023).

We use the test sets for QAMPARI, QUEST and validation sets for ASQA, ELI5.

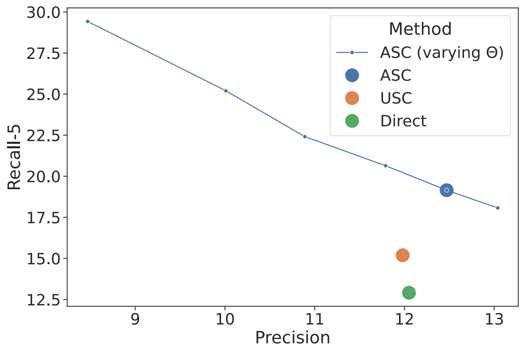

Figure 6: QAMPARI. Increasing Θ improves precision, reduces recall. Adjusting Θ produces preferred answer.

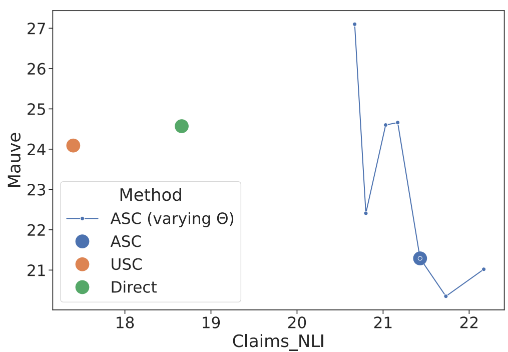

Figure 7: ELI5. Increasing Θ improves QA-F1 and reduces Mauve. Adjusting Θ produces preferred answer.

A.3 Results Continued

We additionally present results from Llama2 in Tables 6 and 7. The trends are exactly similar to ChatGPT, Llama3 described in the main paper.

		ASQA					ELI5			
		#Clusters	**length**	**Mauve**	**Str_EM**	**QA-F1**	**#Clus.**	**length**	**Mauve**	**Claims_Nli**
Llama2	Direct	-	41.88	68	28.71	23.58	-	84.38	46.59	13.98
	ACF		25.78	63.79	28.48	24.73		58.20	38.22	13.70
	FCF		28.71	68.22	28.38	24.64		66.96	35.20	14.57
	USC		63.7	63.63	33.16	26.42		115.82	35.21	17.70
	ASC-F (Ours)	33.57	108.18	62.68	**39.26**	26.54	83.42	148.30	35.25	18.97
	ASC (Ours)	12.68	91.91	70.52	38.82	**27.16**	14.32	143.07	28.09	**19.40**

Table 6: ASQA, ELI5 results. ASC does the best on QA-F1 and demonstrates strong Str_EM. ASC-F picks a large number of clusters and does well on Str_EM. ASC also demonstrates strong Mauve. ASC, ASC-F achieve best Claims_Nli score on ELI5. Results justify that merging of samples is better than picking one sample.

		QAMPARI						QUEST					
	Method	**#Pred**	**Prec**	**Rec**	**Rec-5**	**F1**	**F1-5**	**#Pred**	**Prec**	**Rec**	**Rec-5**	**F1**	**F1-5**
Llama2	Direct	4.86	13.5	9.25	16.23	10.22	14.47	5.46	6.74	4.16	7.66	4.42	6.7
	ACF	3.17	14.94	7.96	13.84	9.69	13.85	3.48	7.9	3.47	6.34	4.14	6.54
	FCF	3.88	14.1	8.93	15.36	10.15	14.22	3.43	8.06	3.78	6.75	4.38	6.77
	USC	7.44	14.07	11.61	20.04	11.64	15.99	9.36	7.76	5.4	10.16	5.38	**7.96**
	ASC-F	27.35	10.74	18.44	29.88	11.52	14.4	28.07	5.63	10.64	19.08	**5.81**	7.67
	ASC	6.08	14.51	12.15	20.58	**12.15**	**16.44**	6.77	7.42	5.52	9.97	5.3	7.86

Table 7: ASC outperforms Direct, USC and ASC-F. ASC-F picks a large number of clusters and does worse on P, F1, F1-5. Results justify that consistency-based cluster selection does better than retrieval-based cluster selection.

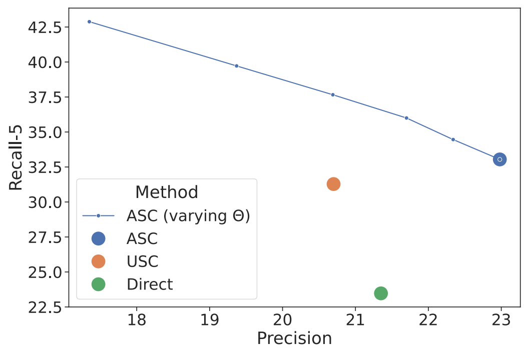

Figure 8: QUEST. Increasing Θ improves precision, reduces recall. Adjusting Θ produces preferred answer.

A.4 Clustering Analysis

In addition to the main results of Table 1, we provide additional results with multiple embedding models and clustering methods. As can be seen in Table 8, the Str_EM performance remains consistent across these variations.

Method	**Emb.**	**Clus.**	**Mauve**	**Str_EM**	**QA-F1**
USC	-	-	40.19	39.05	30.88
ASC	SimCSE	Agglom.	47.01	44.1	32.22
ASC	SimCSE	KMeans	55.25	43.42	31.09
ASC	GTR	Agglom.	53.66	42.61	32.14
ASC	GTR	KMeans	49.85	43.62	32.18

Table 8: Different embedding and clustering methods for ASC. Agglomerative (d=0.15) and Kmeans (K=39)

A.5 Generating oracle Numbers

In both ASQA and QAMPARI, we have access to reference short answers. Evaluation metrics - QA_F1 and precision, recall are all built over these short answers first and then averaged over the entire dataset. For each of these short answers, we find the maximum possible metric value among the 50 generations. This maximum value per short answer is averaged over the entire dataset to get the oracle numbers. For Fig. 2, we use maximum values at the entire response level rather than at a short answer level.

A.6 Prompts

A.6.1 Generation Prompts

We used the exact same generation prompts and few shot exemplars from (Gao et al., 2023) for ASQA, QAMPARI, ELI5. For QUEST (not analysed by (Gao et al., 2023)), we used the same prompt as QAMPARI.

A.7 Summarization Prompt $P_{combine}$

Summarization prompts followed the template shown below. An example for asqa summarization with few shot examples is shown later. We used two shot summarization for both ASQA and ELI5.

Summarization Template

{task instruction}
Question: {....}
Sentence1: {...}
Sentence2: {...}
...
Answer: {....}

{task instruction}
Question: {....}
Sentence1: {...}
Sentence2: {...}
...
Answer: {....}

{task instruction}
Question: { ....}
Sentence1: {...}
Sentence2: {...}
...
Answer:

Summarization Prompt Example

Instruction: You are given an ambiguous question and a few sentences which have some parts of its answer and some irrelevant content. Remove irrelevant sentences and combine all relevant ones i nto a single answer that can address all interpretations of the question. Do not miss any minor details relevant to the question. Also, add any missing details.
Question: Where did bruno live in the boy in the striped pajamas?
Sentence1: Bruno lived in Germany
Sentence2: Bruno moves to Auschwitz when his father got promoted.
Sentence3: He is upset about the move.
Sentence4: Bruno liked playing with his friends.
Sentence5: Bruno lives in Berlin.
Sentence6: Bruno discovered a concentration camp near his new home.
Sentence7: Bruno was an innocent boy.
Answer: Bruno lived in Berlin in Nazi Germany during World War II. His father Ralf gets promoted, and relocates the family to Auschwitz (occupied Poland).

Instruction: You are given an ambiguous question and a few sentences which have some parts of its answer and some irrelevant content. Remove irrelevant sentences and combine all relevant ones i nto a single answer that can address all interpretations of the question. Do not miss any minor details relevant to the question. Also, add any missing details.
Question: Who played nathan on young and the restless? Sentence1: Randy Brooks played nathan on young and restless.
Sentence2: It was played by Lazarre-White in 1994.
Sentence3: He did an amazing job.
Sentence4: Brooks played Nathan.
Sentence5: From 1992, Brooks played nathan.
Sentence6: He was much younger to his predecessors.
Sentence7: He was much younger to his predecessors.
Sentence8: Audience liked nathan's portrayal.
Answer: The role was played by Nathan Purdee from 1984 to 1992. Randy Brooks took over in 1992 but was replaced in 1994 with a younger version of the character, played by Adam Lazarre-White.

Instruction: You are given an ambiguous question and a few sentences which have some parts of its answer and some irrelevant content. Remove irrelevant sentences and combine all relevant ones into a single answer that can address all interpretations of the question. Do not miss any minor details relevant to the question. Also, add any missing details.
Question: What's the marketing strategy of skipping a number in a numbered line of products?

Question	Who has the highest goals in world football?
Cluster 1	• As of August 2021, the soccer player recognized as having the highest number of goals in world football is **Josef Bican**. • As of 2021, the title for the highest goals scorer in world football is held by **Josef Bican** from Austria.
Cluster 2	• As of July 2021, the player with the highest number of goals in world football is the Portuguese forward, **Cristiano Ronaldo**. • As of October 2021, the professional footballer who holds the record for the most career goals in international football is **Cristiano Ronaldo** of Portugal. • As of August 2021, **Cristiano Ronaldo** from Portugal is the player with the highest number of goals in world football.
Cluster 3	• As of September 2021, **Lionel Messi** holds the record for the most career goals in world football, with a total of 740 goals in 943 games for club and country. • The name that currently holds the title of having the most official goals scored in world football by a male player is **Lionel Messi** of Argentina, with a total of 756 goals scored as of January 2021. • As of August 2021, **Messi** has scored a total of 744 goals in his professional career, surpassing previous record-holder Pele's 767 career goals. • As of May 2021, **Lionel Messi** holds the record for the most goals scored in world football with a total of 673 goals in his career.
Cluster 4	• However, there are differing opinions and **methods of calculating** all-time goals in soccer, so the number of goals scored by individual players may vary depending on the criteria used. • However, it is worth noting that determining the "highest goals" can be subjective due to variations in scoring records and **counting methods** across different leagues and competitions.
Cluster 5	• In club football, the player with the highest number of goals is **Lionel Messi** of Argentina, who holds the record for the most goals scored for a single club, with 682 goals for Barcelona. • **Lionel Messi**, the Argentine forward for FC Barcelona, currently holds the record for the most goals scored in world football. • The player with the highest number of goals in world football is currently **Lionel Messi**, an Argentine professional footballer who plays for FC Barcelona and the Argentine national team.

Table 9: Qualitative Analysis - Clustering.