MiniCheck: Efficient Fact-Checking of LLMs on Grounding Documents

Liyan Tang[◇] **Philippe Laban**[♠] **Greg Durrett**[◇]
[◇]The University of Texas at Austin [♠]Salesforce AI Research
lytang@utexas.edu

Abstract

Recognizing if LLM output can be grounded in evidence is central to many tasks in NLP: retrieval-augmented generation, summarization, document-grounded dialogue, and more. Current approaches to this kind of fact-checking are based on verifying each piece of a model generation against potential evidence using an LLM. However, this process can be very computationally expensive, requiring many calls to a model to check a single response. In this work, we show how to build small fact-checking models that have GPT-4-level performance but for 400x lower cost. We do this by constructing synthetic training data with GPT-4, which involves creating realistic yet challenging instances of factual errors via a structured generation procedure. Training on this data teaches models to check each fact in the claim and recognize synthesis of information across sentences. For evaluation, we unify datasets from recent work on fact-checking and grounding LLM generations into a new benchmark, LLM-AGGREFACT. Our best system MiniCheck-FT5 (770M parameters) outperforms all systems of comparable size and reaches GPT-4 accuracy. We release LLM-AGGREFACT, code for data synthesis, and models.[1]

1 Introduction

Freeform generation of responses is a flexible way to employ large language models (LLMs) for question answering, summarization, and beyond. However, this kind of generation can lead to factual errors, the "hallucination" problem in LLMs (Falke et al., 2019; Maynez et al., 2020; McKenna et al., 2023; Zhang et al., 2023a). Such errors arise in generation settings where an LLM is prompted closed-book, but its parametric knowledge may be insufficient to produce the right facts (Min et al., 2023; Mallen et al., 2023; Zhou et al., 2023; Chen et al., 2023). Different but related errors occur in grounded generation settings where evidence is already available, like summarization of input documents or retrieval-augmented question answering, where an LLM can blend information incorrectly (Liu et al., 2023; Tang et al., 2024).

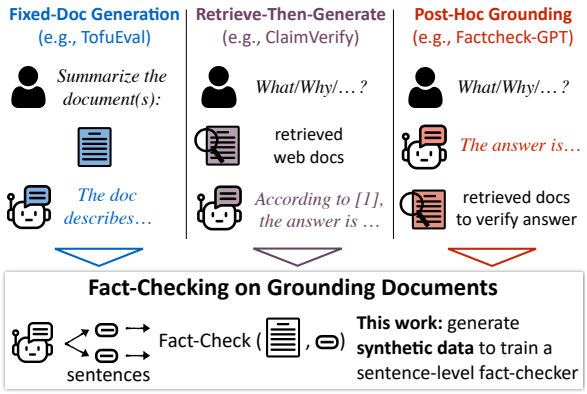

Figure 1: We unify the task of fact-checking across various settings that rely on grounding documents. We train a small sentence-level fact-checker by leveraging new synthetically generated data, which demonstrates strong performance on a new unified benchmark LLM-AGGREFACT, comparable to GPT-4 but 400x cheaper.

Past work has largely dealt with these problems separately. We can post-hoc verify closed-book generated answers by retrieving supporting documents and checking the answer against them (Gao et al., 2023; Malaviya et al., 2024; Jacovi et al., 2024), which requires precise checking as many statements are not exactly supported or may have conflicting information available (Wang et al., 2023; Glockner et al., 2024). When the grounding is known in settings like summarization, the attribution problem can be cleanly framed as document-level textual entailment (Nie et al., 2020a; Yin et al., 2021) and has been studied extensively for smaller language models (Falke et al., 2019; Goyal and Durrett, 2020, 2021; Laban et al., 2022; Tang et al.,

[1]https://github.com/Liyan06/MiniCheck; see latest leaderboard at llm-aggrefact.github.io.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 8818–8847
November 12-16, 2024

2023a).

The problems in this space have a shared primitive operation: the need to check a statement against grounding documents, either retrieval-augmented content or post-hoc retrieved evidence. We call this primitive **fact-checking on grounding documents**, shown in Figure 1. Implementations of this primitive need to be accurate, spotting subtle errors while maintaining a low false positive rate, as most generated statements are correct. They also need to be efficient: a single LLM response may contain dozens of facts to verify, and self-verification with an LLM may increase cost by an order of magnitude (Weng et al., 2023; Gero et al., 2023). For instance, the 110-150 word biographies in FActScore (Min et al., 2023) contain 26-41 atomic facts that are checked against 5 documents each, resulting in 130-205 entailment checks.

In this work, we build an efficient system for fact-checking on grounding documents. Our key insight is to develop a new synthetic training dataset which is tailored to the complexities of the fact-checking task. Unlike standard distillation from LLMs (Taori et al., 2023; Hsieh et al., 2023), this setting differs in that we do not necessary have access to task instances that we can label with strong LLMs. For instance, in datasets like ExpertQA (Malaviya et al., 2024), even the inputs to the LLM are expert-written questions. As a result, we synthesize challenging fact-checking instances from the ground up, as a scalable way to teach a small model how to simultaneously verify multiple facts in a sentence against multiple sentences in grounding documents. Our system, MiniCheck, is an instance of Flan-T5 (Chung et al., 2022) fine-tuned on this data plus standard entailment data (Nie et al., 2020a).

For our experiments, we introduce a new unified benchmark, LLM-AGGREFACT, which aggregates 10 existing datasets for both closed-book and grounded generation settings. In each constituent dataset, sentence-level factual errors are labeled by human annotators. We show that MiniCheck can perform as well as GPT-4 in aggregate and substantially outperform past fine-tuned systems like AlignScore (Zha et al., 2023). Moreover, we find that decomposition of sentences into atomic facts, which has been explored in past work (Kamoi et al., 2023; Gao et al., 2023; Wang et al., 2023), is not necessary to achieve this high performance.

Our contributions are as follows: (1) Two synthetic data generation methods to address the challenges of fact-checking on grounding documents. (2) A new benchmark unifying factual evaluation on closed-book and grounded generation settings. (3) Evaluation shows that our MiniCheck system can beat previous specialized systems by 4% to 10% in absolute values, despite using less fine-tuning data, and is on par with GPT-4 with a much smaller model size, faster inference speed, and 400 times less cost. Furthermore, we can do this without a separate claim decomposition step.

2 Background and Motivation

Problem Setup: Claim Verification We assume a collection of statements to be checked consisting of sentences $\mathbf{c} = [c_1, \ldots, c_{|\mathbf{c}|}]$. Typically, this will be a sequence of sentences produced by an LLM. Each sentence c_i has an associated set of grounding documents $\mathcal{D}_i = \{D_{i,1}, \ldots, D_{i,|\mathcal{D}_i|}\}$. These different $\mathcal{D}_i$ per sentence accommodate post-hoc retrieval settings where each sentence has different retrieved evidence; however, some settings may use shared evidence across all sentences or even a single grounding document for tasks like single-document summarization (i.e., all $\mathcal{D}_i$ only contain the document being summarized).

We view these sentences as *claims*. Our goal in this work is to build a system that can validate each claim against the documents. Following Laban et al. (2022), we define a discriminator

$$M(D_{i,j}, c_i) \in \{0, 1\},$$

that classifies each claim c_i into `unsupported`, 0, or `supported`, 1, according to a provided document $D_{i,j}$.[2]

This process makes two assumptions. First, we assume that **each supported claim can be validated against a single document**; that is, claims are "atomic enough". Our methodology can be generalized to handle claims supported by multiple documents by simply appending multiple documents in the context of M, but we did not find it necessary in any of the datasets we studied.

Second, we assume that **we can perform our entailment checks on each sentence c_i on its own, without context $c_{<i}$.** In general, sentences do need context to be understood (e.g., most sentences starting with pronouns), but this can be resolved through the use of a *decontextualization* step (Choi et al., 2021). Section 7 examines whether such a step improves performance of our system.

[2]Following past work (Kamoi et al., 2023; Sanyal et al., 2024), we disregard the usual "contradiction" class from textual entailment, as contradictions are rare in our benchmark.

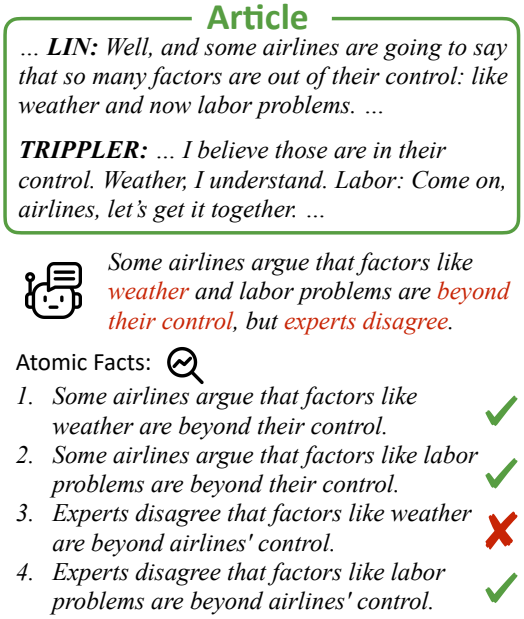

Figure 2: An example dialogue snippet with an LLM-generated summary sentence, from the TofuEval dataset.

We judge a sentence c_i by taking $\max_j M(D_{i,j}, c_i)$: it is supported if and only if there exists some document that supports it.

Challenges of verification Two aspects of the task make this process challenging. First, there may be several individual facts in $D_{i,j}$ which are necessary to validate a claim c_i. For example, the LLM-generated sentence in Figure 2 can be broken down into four atomic facts. Each fact must be checked *even if they are not explicitly materialized.*

Second, and relatedly, the claim, or a fact in the claim, may require making inferences that span multiple sentences within $D_{i,j}$. In Figure 2, Lin initially argues that airlines have no control over either weather or labor issues. However, Trippler's later statement, *"Weather, I understand. Labor: Come on, airlines, let's get it together,"*, implies agreement that weather is uncontrollable but suggests that labor problems are within the airlines' control. This indicates that the third atomic fact is unsupported by the document.

We argue that existing specialized fact-checkers fall short in effectively considering all atomic facts within a claim to be verified and struggle with reasoning across multiple sentences. Results in Appendix A.1 support this characterization. To address this issue, we come up with two synthetic data generation methods (Section 3) to enhance the models' ability in these areas. We discuss the relation to prior work in Section 8.

3 Methodology: Training Data Synthesis

To address these challenges, new data is required. Existing datasets like MNLI (Williams et al., 2018) and ANLI (Nie et al., 2020a) do not feature instances that reflect the complexity of LLM fact-checking. Annotation of real errors is challenging to scale; datasets of such errors (including those in LLM-AGGREFACT) are largely test-only.

Our goal is to construct a dataset $\{(D_i, c_i, y_i)\}_{i=1}^N$ of N instances of documents D_i paired with claims c_i with label $y_i \in \{0, 1\}$, using two novel synthetic data generation methods (Figure 3). Statistics about our final synthetic training data can be found in Table 1. A small-scale human evaluation of the synthetic data quality can be found in Appendix A.3. Additional details, including the sources of claims and documents and examples of generated data, can be found in Appendix D. We provide all prompts and quality assurance details in Appendix H.

3.1 Claim to Doc (C2D) Generation

In the C2D method, we assume that we have access to a set of human-written claim statements. The goal is to generate synthetic documents that require models be able to check multiple facts in the claim against multiple sentences each.

Step 1: Claim decomposition Given a claim c, we first decompose it into a set of atomic facts $\mathbf{a}$ with GPT-3.5: $\texttt{Decomp}(c) = \{a_1, \ldots, a_l\}$.

Step 2: Atomic fact expansion For the claim c, we ask GPT-4 to generate a pair of sentences for each of its atomic facts with a 4-shot prompt:

$$\texttt{SentPair}(a_i) = (s_{i,1}, s_{i,2}), \forall i \in \{1, \ldots, l\}.$$

The generated sentence pairs are designed such that the atomic fact is supported if and only if the information from both sentences is combined.

Step 3: Supporting document generation After expanding atomic facts $\mathbf{a}$ into sentences $\mathbf{s} = \{s_{1,1}, s_{1,2}, \ldots, s_{l,1}, s_{l,2}\}$, we ask GPT-4 to generate a document D that mentions all sentences from the generated sentence pairs in its own words $D = \texttt{PassageGen}(\mathbf{s})$ with a zero-shot prompt.[3]

By following these steps, we create a triplet $(D, c, y = 1)$. This procedure increases the diffi-

[3]We ask GPT-4 to not state deduced facts or conclusions based on the provided sentences, and we find that GPT-4 can follow this instruction well.

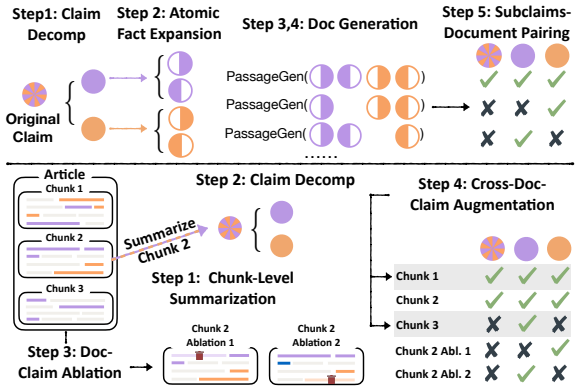

Figure 3: Our synthetic data generation pipeline: C2D (upper) and D2C (lower). We illustrate with a claim that contains two atomic facts. Examples of generated data can be found in Appendix D.

culty of the task by ensuring that multiple-sentence reasoning is required to correctly classify a claim.

Step 4: Nonsupporting document generation By construction, an atomic fact a_i in the claim c is supported by the sentence pair $(s_{i,1}, s_{i,2})$ mentioned in the generated document D. Therefore, by omitting one of the sentences from the pair in a newly generated document D', it is likely that a_i, and consequently c, is no longer supported by D' (except in cases of redundancy in the sentences $\mathbf{s}$). More formally, we can construct a document $D'_{a_{i\setminus j}}$ that *probably* cannot support fact a_i in c (and hence c) by removing sentence $s_{i,j}$ from its sentence pair:

$$D'_{a_{i\setminus j}} = \texttt{PassageGen}(\mathbf{s} \setminus s_{i,j}),$$

for all $i \in \{1, \ldots, l\}$ and $j \in \{1, 2\}$ (Figure 3; top right). To collect documents that do not support the claim c, we retain $D'_{a_{i\setminus j}}$ if a_i cannot be supported by the information combined from the remaining sentence from its sentence pair and other atomic facts $(s_{i,3-j} \cup \{\mathbf{a} \setminus a_i\})$ via an entailment check by GPT-4. Note that this entailment check is again more accurate than directly checking a_i against $D'_{a_{i\setminus j}}$ due to the shorter context.

Step 5: Pairing subclaims and generated documents We have collected tuples $(D, c, 1)$ and $(D'_{a_{i\setminus j}}, c, 0)$ for some i and j. We can further augment this data to produce more examples. We first generate a power set $\texttt{Power}(\mathbf{a})$, that consists of all possible subsets of atomic facts $\mathbf{a}$ in c, but excludes the empty set. We then create a set of augmented subclaims $\text{Aug}(c)$ by merging atomic facts from each subset:

$$\text{Aug}(c) = \{\texttt{Merge}(\mathbf{a}') : \forall \mathbf{a}' \in \texttt{Power}(\mathbf{a})\}.$$

It follows that we obtain tuples $(D, c', 1)$ for every $c' \in \text{Aug}(c)$. Similarly, for each $D'_{a_{i\setminus j}}$, we generate tuples $(D'_{a_{i\setminus j}}, \texttt{Merge}(\mathbf{a}'), 1)$ if $a_i \notin \mathbf{a}'$, indicating that the document still supports the subclaim absent the atomic fact a_i. Conversely, we have $(D'_{a_{i\setminus j}}, \texttt{Merge}(\mathbf{a}'), 0)$ if $a_i \in \mathbf{a}'$, suggesting that the document does not support the subclaim due to the absence of a_i.

Because the same subclaim is supported by certain documents and unsupported by others depending on the presence or absence of specific atomic facts, we achieve the same benefits that training on contrast sets provides (Cao and Wang, 2021; Liu et al., 2022; Tang et al., 2023b), namely making the model more sensitive to the specifics of the decision boundary and encouraging it to consider all atomic facts within a claim during prediction.

3.2 Doc to Claim (D2C) Generation

In the D2C method, our objective is to enhance the diversity of documents and ensure that the documents are more realistic than those in C2D, thereby reducing the distribution shift between synthetic documents used during training and real documents at test time. To achieve this, we assume that we have access to a set of human-written documents to start with. The goal is to generate claims and pair them with portions of these human written documents, which, once again, require multi-sentence, multi-fact reasoning to check the claims.

Step 1: Chunk-level summarization We first divide a human written document into three chunks $\{D_1, D_2, D_3\}$ with approximately equal length. We then use GPT-4 to generate a summary sentence for each chunk, resulting in a set of summary sentences $\mathbf{c} = \{c_1, c_2, c_3\}$. We assume these generated summary sentences are factually consistent with respect to their corresponding chunks, *i.e.* $(D_i, c_i, 1)$ for all i, as each chunk is short and LLMs can almost always generate factual summaries in this setting (Zhang et al., 2024).

Step 2: Claim decomposition and subclaim augmentation Similar to the C2D method, for a summary sentence c_i in $\mathbf{c}$, we decompose it into atomic facts $\mathbf{a_i} = \{a_{i,1}, \ldots, a_{i,l}\}$, and create a set of augmented subclaims $\text{Aug}(c_i) = \{\texttt{Merge}(\mathbf{a}'_i) : \forall \mathbf{a}'_i \in \texttt{Power}(\mathbf{a}_i)\}$.

Step 3: Document-claim augmentation This step aims to do data augmentation on a (D_i, c_i) pair. Given a chunk $D_i = \texttt{Concat}(\mathbf{s})$, which is the

Data	Size	Uniq. Claim	Uniq. Doc	Doc Len	Claim Len	% of Neg
C2D	7076	2004	1188	189	19	42%
D2C	7319	1392	4967	164	12	65%

Table 1: **Statistics of synthetic training data.** Amount of synthetic data for training, the number of unique claims and documents, the average number of words in documents and claims, and the proportion of unsupported claims.

concatenation of n sentences $\mathbf{s} = \{s_{i,1}, ..., s_{i,n}\}$, we construct new documents by iteratively removing each sentence $s_{i,j}$ from $\mathbf{s}$:

$$D'_{i\setminus j} = \texttt{Concat}(\mathbf{s} \setminus \{s_{i,j}\}).$$

We then determine the entailment label for each atomic fact $a_{i,k}$ in c_i, where $k \in \{1, \ldots, l\}$:

$$L^{-j}(a_{i,k}) = \text{Ent}(D'_{i\setminus j}, a_{i,k}) \in \{0, 1\}.$$

Similar to step 5 in C2D, if $L^{-j}(a_{i,k}) = 1$ for all $a_{i,k} \in \mathbf{a}'_i$, we create tuples $(D'_{i\setminus j}, \texttt{Merge}(\mathbf{a}'_i), 1)$. Conversely, if there exists any $a_{i,k} \in \mathbf{a}'_i$ such that $L^{-j}(a_{i,k}) = 0$, we then create tuples $(D'_{i\setminus j}, \texttt{Merge}(\mathbf{a}'_i), 0)$.

Step 4: Cross-document-claim augmentation The objective of this step is to perform data augmentation on a (D_j, c_i) pair, where $j \neq i$. The rationale behind this is that the important information in a document can be conveyed multiple times in various ways. Given that each chunk D_i has an associated summary c_i, it is probable that the summary c_i conveys some information that can be indirectly supported by other chunks D_j within the document, even if D_j are not used to generate c_i. Therefore, we consider chunks D_j, where $j \neq i$, as more challenging chunks to either support or refute the claim c_i or its atomic facts $\mathbf{a}_i$.

More formally, we determine the entailment label for each atomic fact $a_{i,k}$ in c_i, using the document chunk D_j, where $k \in \{1, \ldots, l\}$, and $j \neq i$:

$$L^{D_j}(a_{i,k}) = \text{Ent}(D_j, a_{i,k}) \in \{0, 1\}.$$

If $L^{D_j}(a_{i,k}) = 1$ for all $a_{i,k} \in \mathbf{a}'_i$, we create tuples $(D_j, \texttt{Merge}(\mathbf{a}'_i), 1)$. Conversely, if there exists any $a_{i,k} \in \mathbf{a}'_i$ such that $L^{D_j}(a_{i,k}) = 0$, we then create tuples $(D_j, \texttt{Merge}(\mathbf{a}'_i), 0)$.

3.3 MiniCheck Models

We fine-tune three models with various model architectures by leveraging our synthetic data. We use the standard cross-entropy loss for all models. See Appendix G for training details.

MiniCheck-DBTA and MiniCheck-FT5 As models trained on the ANLI dataset (Nie et al., 2020a) have demonstrated strong performance (Kamoi et al., 2023; Honovich et al., 2022), we integrate our data with the ANLI dataset for fine-tuning `deberta-v3-large` (He et al., 2021) and `flan-t5-large` (Chung et al., 2022). We take a subset (21K) of the ANLI training data, selecting examples where their trained entailment models made incorrect predictions during dataset construction. Training on more of ANLI was not effective.

Combining these 21K datapoints with our 14K-sized dataset, we have 35K training datapoints in total. We map the labels `contradiction` and `neutral` from ANLI to `unsupported`.

MiniCheck-RBTA We also explore whether it is possible to improve upon the previous AlignScore (Zha et al., 2023) system, the existing SOTA specialized fact-checking model. We fine-tune the tuned `roberta-large` (Liu et al., 2019) model from AlignScore with a binary classification head, on our 14K synthetic datapoints.

Producing classification decisions Although our task is framed as binary classification, in reality the models we have are of the form $M(D_i, c_i) \rightarrow z \in \mathbb{R}$, mapping each (document, claim) pair to a score in the range $z \in [v_{\min}, v_{\max}]$. Following Laban et al. (2022); Zha et al. (2023); Tang et al. (2023a), we convert each method into a binary classifier $M(D_i, c_i) \rightarrow \{0, 1\}$ by picking a threshold t such that we predict 1 if $M(D_i, c_i) > t$ and 0 otherwise. Unless otherwise specified, we set $t = 0.5$.

4 LLM-AGGREFACT Benchmark

We construct a fact verification benchmark, LLM-AGGREFACT, by aggregating 10 of the most up-to-date publicly available datasets on factual consistency evaluation across both closed-book and grounded generation settings.

Characteristics In LLM-AGGREFACT, all datasets contain human-annotated `(document, claim, label)` tuples. The documents come from diverse sources, including Wikipedia paragraphs, interviews, web text, covering domains such as news, dialogue, science, and healthcare. The claims to be verified are mostly generated from recent generative models (except for one dataset

Type	Dataset	Feature
Fixed-Doc Generation	AGGREFACT (CNN/XSum)	Summaries from SOTA fine-tuned summarizers
	TOFUEVAL (MediaS/MeetB)	Topic-focused dialogue summaries from LLMs
Retrieve-then Generate	CLAIMVERIFY	(Check-worthy) sentences from LLMs/ search engines' responses to search queries
	LFQA	
	EXPERTQA	
Post-Hoc Grounding	EXPERTQA	
	REVEAL	
	FACTCHECK-GPT	
Written Claims	WICE	Wikipedia claims with citations

Figure 4: 10 datasets in LLM-AGGREFACT. Details of these datasets as well as related but excluded datasets can be found in Appendix C.

of human-written claims), without any human intervention in any format, such as injecting certain error types into model-generated claims. An overview of the LLM-AGGREFACT is shown in Figure 4, with statistics and detailed dataset descriptions in Appendix C.

4.1 Benchmark Details

Validation/Test set split For the datasets from AGGREFACT and TOFUEVAL, as well as WICE and CLAIMVERIFY, we directly use the existing validation and test splits from the original work. For REVEAL, FACTCHECK-GPT, EXPERTQA and LFQA, we randomly divide each of them into validation and test sets (50%/50%), assuring that responses to unique queries do not appear in both sets.

One potential use of the validation data is to allow for **per-dataset threshold tuning**. This setting is used in substantial past work (Laban et al., 2022; Luo et al., 2023; Zha et al., 2023; Tang et al., 2023a, 2024). However, we do not follow this trend in order to focus on building systems that can be deployed zero-shot across multiple downstream tasks, without any additional hyperparameter tuning. Instead, for $M(d, c) \rightarrow z \in [v_{\text{min}}, v_{\text{max}}]$, the threshold is set as the midpoint of the output score range $t = (v_{\text{min}} + v_{\text{max}})/2$, which is 0.5 for most fact-checkers. In practice, most fact-checkers return scores at the extremes of the range, so small tweaks on this procedure have little effect. See Appendix B.1 for the results of the threshold tuning setting, which yields qualitatively similar results.

Evaluation Metric Following Laban et al. (2022); Fabbri et al. (2022); Tang et al. (2023a), we evaluate the performance of fact-checkers using balanced accuracy (BAcc): $\text{BAcc} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP+FN}} + \frac{\text{TN}}{\text{TN+FP}}\right)$, where TP, TN, FP, and FN represent true/false positives/negatives.

5 Experimental Setup

We include the following specialized fact-checkers: **T5-NLI-Mixed** (Honovich et al., 2022), **DAE** (Goyal and Durrett, 2021), **QAFactEval** (Fabbri et al., 2022), **SummaC-ZS** and **SummaC-CV** (Laban et al., 2022), **AlignScore** (Zha et al., 2023), and **FT5-ANLI-L** that fine-tunes `flan-t5-large` on the full ANLI training set. A meta-comparison of those specialized fact-checkers and our models can be found in Table 3. More inference details can be found in Appendix E.2.

We also include the following LLMs as fact-checkers: **Gemini-Pro** (Team et al., 2023), **PaLM2-Bison** (Thoppilan et al., 2022), **Mistral-8x7B**, **Mistral-Large** (Jiang et al., 2024), **Claude 2.1**, **Claude 3 Opus** (Bai et al., 2022), **GPT-3.5** and **GPT-4** (OpenAI, 2023). More details about the models can be found in Appendix E.1. For the LLM-based fact-checkers, we adapt a prompt from Luo et al. (2023) for zero-shot prediction, which can be found in Appendix H.

6 Results

6.1 Main Results

Our synthetic data improves performance across diverse model architectures. Table 2 demonstrates that our synthetic data gives strong performance when used in three different backbone models: RoBERTa, DeBERTa, and Flan-T5. These models outperform prior models of a similar scale. Notably, MiniCheck-FT5 achieves a 4.3% overall improvement over AlignScore, outperforming it on 6 out of 10 datasets and matching its performance on the remaining 4. We attribute its additional 2% gain over MiniCheck-RBTA and -DBTA to its larger model size. However, model size alone does not guarantee superior performance, as evidenced by T5-NLI-Mixed and FT5-ANLI-L, which, despite being trained on NLI datasets, underperform on most of the benchmark settings. This underscores the importance of training data selection in addition to model capacity.

Our models achieve performance on par with the most capable LLM-based fact-checkers. In

Model Name	LLM-AGGREFACT (*without* threshold tuning)										
	AGGREFACT		TOFUEVAL		WICE	REVEAL	CLAIM VERIFY	FACT CHECK	EXPERT QA	LFQA	Avg
	CNN	XSum	MediaS	MeetB							
Gemini-Pro	49.4	60.6	63.8	65.8	65.8	85.5	61.8	76.8	56.8	75.9	66.2
PaLM2-Bison	52.4	59.0	68.3	73.6	63.4	84.2	60.5	76.4	56.6	71.4	66.6
Mistral-8x7B	55.0	65.5	68.5	73.3	63.8	80.8	64.3	75.1	56.3	70.8	67.3
GPT-3.5	63.2	72.4	66.8	73.4	68.5	84.7	65.2	70.8	57.2	73.8	69.6
Claude-2.1	59.9	66.4	69.2	72.3	64.3	88.2	69.7	79.3	59.8	78.2	70.7
Mistral-Large	58.4	76.3	67.3	78.9	76.6	88.4	67.6	79.0	60.0	81.7	73.4
Claude-3 Opus	65.2	72.4	74.1	82.4	75.0	83.8	69.3	78.8	58.8	81.6	74.1
GPT-4	66.7	76.5	71.4	79.9	80.4	87.8	67.6	79.9	59.2	83.1	75.3
SummaC-CV	65.2	54.5	63.7	62.8	54.3	67.7	70.9	53.4	54.9	62.1	62.1
T5-NLI-Mixed	54.6	52.3	59.1	55.3	55.3	87.2	59.5	69.0	55.6	61.8	61.0
FT5-ANLI-L	51.2	60.0	57.4	60.1	67.0	77.5	58.3	67.7	52.2	63.0	61.4
DAE	50.8	59.1	65.1	69.5	58.5	81.3	64.0	72.5	56.2	72.2	64.9
QAFactEval	54.3	62.1	61.3	65.7	62.5	83.2	73.2	66.1	56.0	80.6	66.5
SummaC-ZS	51.1	61.5	69.5	71.0	62.8	85.3	69.7	75.2	55.2	77.6	67.9
AlignScore	52.4	71.4	69.2	72.6	66.0	85.3	69.6	74.3	58.3	84.5	70.4
MiniCheck-DBTA	64.2	71.0	69.3	72.7	69.4	87.3	75.6	73.0	58.9	83.9	72.6
MiniCheck-RBTA	63.7	70.8	71.9	75.9	67.6	88.8	77.4	73.3	57.4	84.4	72.7
MiniCheck-FT5	69.9	74.3	73.6	77.3	72.2	86.2	74.6	74.7	59.0	85.2	74.7

Table 2: Performance (BAcc) of models on the test set of LLM-AGGREFACT without per-dataset threshold tuning. Models are split into *LLM-based fact-checkers* | *specialized fact-checkers* | *Ours*. We highlight the best performance for each dataset, where multiple green highlights indicate systems indistinguishable from the best according to a paired bootstrap test with 1000 runs and p-value < 0.05. Details for -Dectx and -Decmp are in Section 7. Our MiniCheck models outperform other specialized evaluators and MiniCheck-FT5 reaches the performance of GPT-4.

Model Name	Backbone Model	Model Size	# FT Data	Cost ($)
T5-NLI-Mixed	T5-XXL	11B	1,697K	7.39
FT5-ANLI-L	Flan-T5-L	770M	163K	0.24
DAE	ELECTRA-B	110M	95K	0.26
QAFactEval	multiple*	1.4B	-	1.87
SummaC-ZS	ALBERT-XL	60M	371K	0.85
SummaC-CV	ALBERT-XL	60M	381K	0.85
AlignScore	RoBERTa-L	355M	4,700K	0.20
MiniCheck-RBTA	AlignScore	355M	14K	0.20
MiniCheck-DBTA	DeBERTa-L	355M	35K	0.20
MiniCheck-FT5	Flan-T5-L	770M	35K	0.24

Table 3: Comparison of specialized fact-checkers on model sizes, training data sizes, and the inference cost ($0.8/GPU-hr) on the 13K LLM-AGGREFACT test set. *QAFactEval contains several model components, which sum up to 1.4B in size.

the top rows of Table 2, we present the performance of strong LLM-based fact-checkers. We observe that existing specialized fact-checkers achieve similar performance to non-frontier LLM-based fact-checkers like Mistral-8x7B and GPT-3.5. MiniCheck-RBTA and MiniCheck-DBTA can surpass these non-frontier LLM-based fact-checkers by a large margin. MiniCheck-FT5 achieves the same performance as Claude-3 Opus and is close to GPT-4, but with a much smaller model size.

Extended Analysis See Appendix A for an intrinsic evaluation on our synthethic data and an ablation study on our best model MiniCheck-FT5.

6.2 Computational Cost Comparison

We compare the computational cost of specialized fact-checkers and LLMs on the test set of LLM-AGGREFACT. For specialized fact-checkers, we use our own hardware and convert the prediction time on our GPUs to the equivalent cost of using cloud computing services (see Appendix E.2.1 for details). For LLM-based fact-checkers, we compute the costs of corresponding API calls. Results are shown in Table 3 and 4. We see that specialized models in general have much lower inference costs. In particular, our most capable model MiniCheck-FT5 has almost the same performance as GPT-4 but is more than 400 times cheaper.

7 Rethinking LLM Fact-Checking

We now revisit two other stages of the typical LLM fact-checking pipeline: claim decomposition and decontextualization. Surprisingly, we find that claim decomposition is not needed in our settings, contradicting prior work (Yang and Zhu, 2021; Kamoi et al., 2023). Furthermore, we find that decontextualization doesn't help on our benchmark,

Model Name	Cost ($)	Model Name	Cost ($)
Gemini-Pro	5.24	Claude-2.1	89.9
PaLM2-Bison	10.9	Claude-3 Opus	165
Mistral-8x7B	7.78	GPT-3.5	4.75
Mistral-Large	90.2	GPT-4	107
GPT-4-Dectx	161	GPT-4-Decmp	212

Table 4: Inference cost comparison for API models on the 13K LLM-AGGREFACT test set. Both decontextualization and decomposition add cost to GPT-4. Overall, decoding our test set with the most capable models incurs significant cost.

Model	Decomposition	Decontextualization
GPT-4	75.6 (↑ **0.3**)	75.3 (+**0.0**)
SummaC-CV	58.8 (↓ **3.3**)	60.8 (↓ **1.3**)
QAFactEval	64.6 (↓ **1.9**)	66.4 (↓ **0.1**)
SummaC-ZS	69.1 (↑ **1.2**)	67.7 (↓ **0.2**)
AlignScore	71.5 (↑ **1.1**)	70.4 (+**0.0**)
MiniCheck-RBTA	73.2 (↑ **0.5**)	72.4 (↓ **0.3**)
MiniCheck-DBTA	72.7 (↑ **0.1**)	71.2 (↓ **1.4**)
MiniCheck-FT5	73.3 (↓ **1.4**)	74.1 (↓ **0.6**)

Table 5: Average performance on the test set of LLM-AGGREFACT by aggregating predictions on decomposed claims (left); doing claim decontextualization where it is applicable (right). We show the performance change compared to predicting using original claims from Table 2. Full results in Appendix Tables 8 and 9.

although we believe that it is needed in general.

7.1 Claim Decomposition

We also experiment with a setting using claim decomposition. In this setting, we decompose each claim c_i into atomic facts $\mathbf{a}_i$ with the prompt from Kamoi et al. (2023) and use a fact-checker to predict the factuality label for each $(D_i, a_{i,k})$ pair, $k \in \{1, \ldots, l\}$. If all atomic facts are supported by the document, then the claim is supported, and unsupported otherwise. We do this for every dataset except FactCheck-GPT which is already atomic facts. There are typically 2-4 atomic facts per claim across datasets.

We show the results from GPT-4 and a subset of specialized fact-checkers in Table 5. We observe near-zero performance change for GPT-4 and mixed changes for specialized fact-checkers. **Overall, there is no clear indication that decomposing claims into atomic facts can consistently improve models' performance.** Because this approach increases the inference time and costs by a factor of 2-4 for different datasets, depending on the average number of atomic facts per claim, we believe it should not be used until it provides a clear accuracy benefit.[4]

7.2 Claim Decontextualization

As mentioned in Section 2, our approach relies on being able to check each sentence in isolation. However, phenomena like coreference and ellipsis may make sentences difficult to ground out of context. We can address this with an explicit *decontextualization* step (Choi et al., 2021; Wang et al., 2023; Jacovi et al., 2024). We experiment with TOFUEVAL-MediaS, TOFUEVAL-MeetB, WICE, REVEAL, CLAIMVERIFY, EXPERTQA and LFQA, which are the datasets in our benchmark where sentences need to be interpreted in context (FACTCHECK is already decontextualizaed). We prompt GPT-4 for decontextualization as shown in Appendix H, using the previous claims or response sentences as context to expand the claim. Respectively, 33%, 33%, 39%, 11%, 35%, 47%, and 57% of the claims from those datasets are changed after decontextualization.

In Table 5, we show the average fact-checking performance when using this decontextualization step (see the prompt in Table 24). **These results suggest that models may make decent guesses about context-dependent content, particularly when the retrieval stage already implicitly enforces shared context between the claim and the grounding documents.** However, for tasks such as retrieval-augmented generation, we believe decontextualization still plays a crucial role in ensuring meaningful document retrieval. Furthermore, as LLMs scale further and their responses get more complex, the level of contextualization they feature may be higher, making this step more necessary.

8 Related Work

Hallucinations in LLMs LLMs are prone to hallucinations across various settings (Huang et al., 2023; Zhang et al., 2023b; Rawte et al., 2023), generating information that cannot be supported by any source. For example, in the closed-book setting, where LLMs rely solely on their parametric knowledge, they may fabricate details when describing biographies or providing Wikipedia entity information (Min et al., 2023; Guan et al., 2023; Mallen

[4]Note that for Factcheck-GPT, retrieval operates over individual atomic facts. Decomposition may still be necessary to *retrieve* the relevant information, but our results show that it may not be necessary for *entailment checks*.

et al., 2023). In retrieval-augmented settings, where models have access to external documents to provide responses to user queries, they may generate supplementary information that is not faithful to the provided documents (Chiesurin et al., 2023; Adlakha et al., 2023; Chen et al., 2024). Even when LLMs are provided with gold documents, such as in text summarization and simplification tasks, they still generate factually inconsistent outputs with diverse error types across different domains (Joseph et al., 2023; Shaib et al., 2023; Tang et al., 2024, 2023c). In this work, we construct a new benchmark dataset, LLM-AGGREFACT, which unifies human-annotated model responses across all settings, and evaluate the performance of existing fact-checkers and our proposed ones on the benchmark in detecting such errors.

Methods in Detecting Hallucinations When documents are directly available for model-generated sentences, such as in text summarization (Falke et al., 2019; Kryscinski et al., 2020; Maynez et al., 2020; Fabbri et al., 2021; Tang et al., 2023a) or retrieval-augmented generation (Liu et al., 2023; Malaviya et al., 2024), the entire claims are directly verified against the source documents. However, in cases where such documents are not readily available, such as in close-book generation, Gao et al. (2023); Min et al. (2023); Wang et al. (2023) decompose each generated sentence into atomic facts and then search for relevant documents to support each atomic fact. Alternatively, Malaviya et al. (2024) directly search for relevant documents for each sentence as a whole.

There are two main approaches to verifying sentences against documents. The first involves training specialized fact-checkers specifically designed for factual consistency evaluation, which are primarily evaluated in the context of summarization (Kryscinski et al., 2020; Fabbri et al., 2022; Goyal and Durrett, 2020; Laban et al., 2022). The second approach leverages LLMs as fact-checkers, particularly for evaluating LLM-generated responses from retrieval-augmented generation and closed-book generation (Min et al., 2023; Wang et al., 2023; Malaviya et al., 2024; Gao et al., 2023). In this work, we bridge the gap between these two approaches by evaluating both specialized fact-checkers and LLM-based fact-checkers across all these settings using our new benchmark, LLM-AGGREFACT. We show that our best model can match GPT-4 performance and perform well in all settings without doing sentence decomposition.

Entailment Datasets Our work contributes a new dataset for training textual entailment models over documents or document-chunks. Most prior entailment datasets have been human-authored (Bowman et al., 2015; Williams et al., 2018), which is known to introduce artifacts (Gururangan et al., 2018), or collected in the wild (Kamoi et al., 2023), which is challenging to scale. Past work has automatically generated contrast sets for NLI (Li et al., 2020). DocNLI (Yin et al., 2021) is a restructure of existing datasets where the length of most training examples cannot fit into the input limit of small models. Our work differs from these in its hand-built, synthetic nature to encourage multi-sentence and multi-fact reasoning, which is important to the task of fact-checking on grounding documents.

9 Conclusion

In this work, we introduce two synthetic data generation methods that address key limitations of specialized fact-checkers by encouraging models to verify each atomic fact within a claim and reason across multiple sentences. We also present LLM-AGGREFACT, a new factual consistency evaluation dataset covering both closed-book and grounded generation settings. A model fine-tuned on our synthetic data outperforms all prior specialized fact-checkers on LLM-AGGREFACT, while being much cheaper than LLM-based fact-checkers.

Limitations

Interpretation Like many other specialized fact-checking models, our models do not reveal their internal decision-making processes, making it challenging to localize errors to particular mismatched spans of a claim or document. There are two ways to alleviate this issue. The first is to perform claim decomposition and check the models' prediction labels on each atomic fact, thereby localizing the error from the original claim. Our models show better performance compared to other specialized fact-checkers when using this claim decomposition method (Table 5), but this method can give greater interpretability. The second approach, which can be a future research direction, is to enable our best model, MiniCheck-FT5, or generative models in general, to provide reliable explanations in addition to the binary predictions. We believe a model can provide reliable explanations only if it can first cor-

rectly identify errors in our binary setting, which was the focus of this work.

Multi-Document Reasoning While our benchmark includes instances that evaluate models' ability to reason across multiple sentences, these datasets do not necessitate reasoning over evidence that is significantly separated or spread across various documents. Future research could focus on evaluating the performance of existing models in such scenarios, creating new labeled datasets of errors to expand our benchmark, and developing better fact-checking models to handle these challenges.

Synthetic Data The effectiveness of our synthetic data is demonstrated by the improved performance when training on it across various model architectures. We find that using a pair of sentences to support a fact is a simple method that yields useful training data for our models. However, there are other possible strategies for how atomic facts or claims could be expanded into multiple sentences. We believe that constructing more complex and higher-quality data could be a future direction not only for this work but also for other related tasks. As LLMs continue to advance, the quality of the synthetic data generated using our approach is also expected to improve.

Language Our models are trained exclusively on English data. Although the backbone model, Flan-T5, is trained on multilingual data, we have not systematically assessed how well our model's performance extends to other languages due to the absence of a human-annotated factual consistency evaluation dataset for LLM-generated outputs in non-English languages. We believe that developing a fact-checker that can perform well across multiple languages is important for future work.

Acknowledgments

We would like to thank Jessy Li for comments on a draft of this work. This work was principally supported by a gift from Amazon as part of the UT-Amazon Science Hub. It was partially supported by NSF CAREER Award IIS-2145280, the NSF AI Institute for Foundations of Machine Learning (IFML), a grant from Open Philanthropy, a grant from the UT Austin Office of the Vice President for Research through the "Creating Connections for National Security Research Grants" program, and Good Systems,[5] a UT Austin Grand Challenge to develop responsible AI technologies.

References

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. 2023. Evaluating correctness and faithfulness of instruction-following models for question answering. *arXiv preprint arXiv:2307.16877*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Meng Cao, Yue Dong, and Jackie Cheung. 2022. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3340–3354, Dublin, Ireland. Association for Computational Linguistics.

Shuyang Cao and Lu Wang. 2021. CLIFF: Contrastive learning for improving faithfulness and factuality in abstractive summarization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6633–6649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hung-Ting Chen, Fangyuan Xu, Shane A Arora, and Eunsol Choi. 2023. Understanding retrieval augmentation for long-form question answering. *arXiv preprint arXiv:2310.12150*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Thirty-Eighth*

[5] https://goodsystems.utexas.edu/

AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada, pages 17754–17762. AAAI Press.

Tongfei Chen, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, and Benjamin Van Durme. 2020. Uncertain natural language inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8772–8779, Online. Association for Computational Linguistics.

Sabrina Chiesurin, Dimitris Dimakopoulos, Marco Antonio Sobrevilla Cabezudo, Arash Eshghi, Ioannis Papaioannou, Verena Rieser, and Ioannis Konstas. 2023. The dangers of trusting stochastic parrots: Faithfulness and trust in open-domain conversational question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 947–959, Toronto, Canada. Association for Computational Linguistics.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making sentences stand-alone. *Transactions of the Association for Computational Linguistics*, 9:447–461.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Tobias Falke, Leonardo F. R. Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Zelalem Gero, Chandan Singh, Hao Cheng, Tristan Naumann, Michel Galley, Jianfeng Gao, and Hoifung Poon. 2023. Self-verification improves few-shot clinical information extraction. In *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*.

Max Glockner, Ieva Staliūnaitė, James Thorne, Gisela Vallejo, Andreas Vlachos, and Iryna Gurevych. 2024. AmbiFC: Fact-Checking Ambiguous Claims with Evidence. *Transactions of the Association for Computational Linguistics*, 12:1–18.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2021. Annotating and modeling fine-grained factuality in summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1449–1462, Online. Association for Computational Linguistics.

Jian Guan, Jesse Dodge, David Wadden, Minlie Huang, and Hao Peng. 2023. Language models hallucinate, but may excel at fact verification. *arXiv preprint arXiv:2310.14564*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 161–175, Dublin, Ireland. Association for Computational Linguistics.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Yebowen Hu, Timothy Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. MeetingBank: A benchmark dataset for meeting summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16409–16423, Toronto, Canada. Association for Computational Linguistics.

Dandan Huang, Leyang Cui, Sen Yang, Guangsheng Bao, Kun Wang, Jun Xie, and Yue Zhang. 2020. What have we achieved on text summarization? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 446–469, Online. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roee Aharoni, and Mor Geva. 2024. A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Sebastian Joseph, Kathryn Kazanas, Keziah Reina, Vishnesh Ramanathan, Wei Xu, Byron Wallace, and Junyi Jessy Li. 2023. Multilingual simplification of medical texts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16662–16692, Singapore. Association for Computational Linguistics.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. WiCE: Real-world entailment for claims in Wikipedia. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7561–7583, Singapore. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Philippe Laban, Wojciech Kryscinski, Divyansh Agarwal, Alexander Fabbri, Caiming Xiong, Shafiq Joty, and Chien-Sheng Wu. 2023. SummEdits: Measuring LLM ability at factual reasoning through the lens of summarization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9662–9676, Singapore. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Chuanrong Li, Lin Shengshuo, Zeyu Liu, Xinyi Wu, Xuhui Zhou, and Shane Steinert-Threlkeld. 2020. Linguistically-informed transformations (LIT): A method for automatically generating contrast sets. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 126–135, Online. Association for Computational Linguistics.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Nelson Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025, Singapore. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. BRIO: Bringing order to abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2890–2903, Dublin, Ireland. Association for Computational Linguistics.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. ChatGPT as a Factual Inconsistency Evaluator for Text Summarization.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. ExpertQA: Expert-curated questions and attributed answers. In *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2758–2774, Singapore. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

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

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020a. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020b. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *ArXiv*, abs/2303.08774.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Fabio Petroni, Samuel Broscheit, Aleksandra Piktus, Patrick Lewis, Gautier Izacard, Lucas Hosseini, Jane Dwivedi-Yu, Maria Lomeli, Timo Schick, Michele Bevilacqua, Pierre-Emmanuel Mazaré, Armand Joulin, Edouard Grave, and Sebastian Riedel. 2023. Improving Wikipedia verifiability with AI. *Nature Machine Intelligence*, 5(10):1142–1148.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*.

Soumya Sanyal, Tianyi Xiao, Jiacheng Liu, Wenya Wang, and Xiang Ren. 2024. Minds versus machines: Rethinking entailment verification with language models. *arXiv preprint arXiv:2402.03686*.

Chantal Shaib, Millicent Li, Sebastian Joseph, Iain Marshall, Junyi Jessy Li, and Byron Wallace. 2023. Summarizing, simplifying, and synthesizing medical evidence using GPT-3 (with varying success). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1387–1407, Toronto, Canada. Association for Computational Linguistics.

Liyan Tang, Tanya Goyal, Alex Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryscinski, Justin Rousseau, and Greg Durrett. 2023a. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11626–11644, Toronto, Canada. Association for Computational Linguistics.

Liyan Tang, Yifan Peng, Yanshan Wang, Ying Ding, Greg Durrett, and Justin Rousseau. 2023b. Less likely brainstorming: Using language models to generate alternative hypotheses. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12532–12555, Toronto, Canada. Association for Computational Linguistics.

Liyan Tang, Igor Shalyminov, Amy Wing mei Wong, Jon Burnsky, Jake W. Vincent, Yu'an Yang, Siffi Singh, Song Feng, Hwanjun Song, Hang Su, Lijia Sun, Yi Zhang, Saab Mansour, and Kathleen McKeown. 2024. TofuEval: Evaluating Hallucinations of LLMs on Topic-Focused Dialogue Summarization.

Liyan Tang, Zhaoyi Sun, Betina Idnay, Jordan G. Nestor, Ali Soroush, Pierre A. Elias, Ziyang Xu, Ying Ding, Greg Durrett, Justin F. Rousseau, Chunhua Weng, and Yifan Peng. 2023c. Evaluating large language models on medical evidence summarization. *npj Digital Medicine*, 6(1).

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. `https://github.com/tatsu-lab/stanford_alpaca`.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, and et al. 2023. Gemini: A family of highly capable multimodal models.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam M. Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, Yaguang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, I. A. Krivokon, Willard James Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Hartz Søraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Díaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, V. O. Kuzmina, Joseph Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Rogers Croak, Ed Huai hsin Chi, and Quoc Le. 2022. LaMDA: Language Models for Dialog Applications. *ArXiv*, abs/2201.08239.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, Isabelle Augenstein, Iryna Gurevych, and Preslav Nakov. 2023. Factcheck-GPT: End-to-End Fine-Grained Document-Level Fact-Checking and Correction of LLM Output. *ArXiv*, abs/2311.09000.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575, Singapore. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Xiaoyu Yang and Xiaodan Zhu. 2021. Exploring decomposition for table-based fact verification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1045–1052, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. DocNLI: A large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023a. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. 2024. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023b. Siren's song in the AI ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2023. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14544–14556, Singapore. Association for Computational Linguistics.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. MediaSum: A large-scale media interview dataset for dialogue summarization. In *Proceedings*

of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5927–5934, Online. Association for Computational Linguistics.

A Additional Analysis and Ablations

A.1 Intrinsic Evaluation on C2D/D2C

Our model achieves strong overall performance, but we would like to have more insight as to whether it actually does well at the types of instances in D2C and C2D. We evaluate the performance of QAFactEval, SummaC-ZS, SummaC-CV, AlignScore, FT5-ANLI-L on our held-out synthetic data of C2D and D2C as a way to understand their ability to reason over multiple sentences and consider multiple atomic facts within a claim. There are 2K held-out instances from C2D and D2C, respectively, in the format of `(document, claim, label)` tuples. Performance is measured by BAcc.

Synthetic data ablations Beside those specialized models, we fine-tune `flan-t5-large` on the training set of C2D (**FT5-C2D**) and D2C (**FT5-D2C**), respectively. We evaluate FT5-C2D on D2C as an out-of-distribution (OOD) evaluation set, and evaluate FT5-D2C on C2D.

To demonstrate the necessity of steps in creating our synthetic data, we also create simplified versions of the synthetic data for both methods, denoted as C2D-SIMP and D2C-SIMP, each comprising 7K training examples, same as in C2D and D2C. For the C2D-SIMP method, we ask GPT-4 to directly generate documents that support and not support a provided claim, with the requirement mentioned in the prompt that the inference on the claim should require reasoning over multiple sentences from a document. For the D2C-SIMP method, we ask GPT-4 to generate summary sentences for Google News articles and come up with unsupported summaries by injecting errors into those summary sentences following the method in SummEdit (Laban et al., 2023). We denote models trained on those simplified synthetic data as **FT5-C2D-S** and **FT5-D2C-S**. More details for creating these synthetic datasets can be found in Appendix F.

Synthetic data needs to be carefully constructed for an fact-checker to work well. Figure 5 shows the in-distribution performance of FT5-C2D and FT5-D2C as optimal performance in each sub-figure, represented by the black dashed lines. We observe that models trained on synthetic data

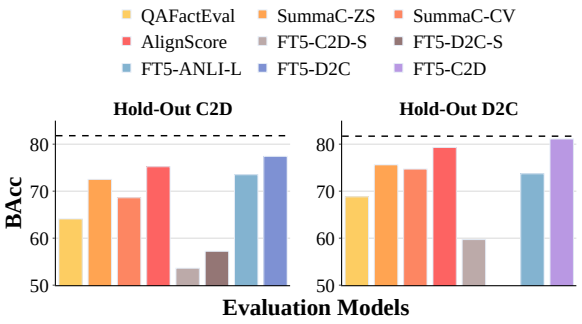

Figure 5: Performance of fact-checkers on the held-outs sets of C2D (left) and D2C (right). The black dashed line shows the in-distribution performance of FT5-C2D (left) and FT5-D2C (right).

with simplified construction steps (C2D-SIMP and D2C-SIMP) fail to develop the desired properties we expect. FT5-D2C-S performs even worse than random chance on the held-out set of C2D. In contrast, models trained on C2D and D2C outperform all other fact-checkers on the OOD held-out set of D2C and C2D, respectively. Additionally, we note that the model trained on 163K ANLI data points fail to reach the performance of models trained solely on 7K synthetic data. Our synthetic data generation methods can effectively encourage models to pay more attention to multiple atomic facts and reason over multiple sentences, even with a limited amount of training data.

A.2 Ablation of D2C/C2D

We observe that there are still gaps between the optimal performance and that achieved by other specialized fact-checkers in Figure 5. Notably, AlignScore demonstrates the best performance among the four specialized metrics, but its performance can still be outperformed by FT5-C2D and FT5-D2C. As shown in Table 2, we can improve AlignScore's performance on the benchmark by fine-tuning it on this small amount of data, effectively equipping it with the desired properties.

To further investigate, we conducted an ablation study on our top-performing model, MiniCheck-FT5, by removing our synthetic data from the training set. The results, presented in Table 6, reveal that the two types of synthetic data complement each other. Notably, the model performs poorly when trained solely on the ANLI subset, with a performance drop of nearly 10% in the absence of threshold tuning. However, the addition of either 7K C2D or 7K D2C to the training data significantly enhances the model's performance and

Model Name	LLM-AGGREFACT (*without* threshold tuning)										Avg
	AGGREFACT		TOFUEVAL		WICE	REVEAL	CLAIM VERIFY	FACT CHECK	EXPERT QA	LFQA	
	CNN	XSum	MediaS	MeetB							
MiniCheck-FT5	69.9	74.3	73.6	77.3	72.2	86.2	74.6	74.7	59.0	85.2	74.7
- C2D	64.7	68.6	70.7	76.6	75.5	85.4	70.4	75.1	58.6	82.0	72.7 (↓ **2.0**)
- D2C	62.9	70.6	70.1	75.9	74.0	83.1	67.1	75.5	58.0	77.9	71.5 (↓ **3.2**)
- BOTH	54.7	59.4	54.1	55.6	61.5	77.1	57.4	65.0	51.9	63.0	59.9 (↓ **14.8**)

Table 6: **Ablation study on the training data.** Models are evaluated on the test set of LLM-AGGREFACT without threshold tuning. We show the average performance downgrade in red. -BOTH drops from 69.1 to 59.9 without threshold tuning.

Model Name	LLM-AGGREFACT (*with* threshold tuning)										Avg
	AGGREFACT		TOFUEVAL		WICE	REVEAL	CLAIM VERIFY	FACT CHECK	EXPERT QA	LFQA	
	CNN	XSum	MediaS	MeetB							
T5-NLI-Mixed	59.9	56.1	60.6	55.9	58.0	87.6	61.5	69.3	56.8	62.8	62.9
DAE	58.6	67.1	67.6	63.2	57.1	83.8	71.1	72.6	58.6	68.5	68.5
QAFactEval	63.9	63.7	64.0	66.8	65.8	85.3	73.3	72.1	57.6	81.5	69.4
SummaC-ZS	63.0	67.2	69.5	70.0	61.8	86.4	69.9	75.7	57.5	82.0	70.4
SummaC-CV	67.6	69.7	68.2	71.0	66.1	87.3	71.3	74.3	59.3	71.3	71.3
AlignScore	62.6	69.6	71.6	71.8	66.8	85.3	72.9	76.5	59.2	85.6	72.2
MiniCheck-RBTA	64.6	70.2	71.1	75.2	73.7	88.0	77.1	77.3	58.4	84.2	73.7
MiniCheck-DBTA	63.4	74.7	69.1	72.8	76.3	87.4	75.5	76.0	58.8	84.1	73.8
MiniCheck-FT5	71.5	74.8	73.7	76.7	75.0	86.4	73.8	76.4	58.6	84.4	75.1

Table 7: **Performance of models on the test set of LLM-AGGREFACT *with* threshold tuning on the validation set.** Balanced accuracy is computed for each model on the 10 datasets in LLM-AGGREFACT, and the average is computed. In each dataset, a factuality metric selects a threshold based on the performance on the corresponding validation set.

robustness.

A.3 Human Evaluation of Synthetic Data

While we use automatic entailment checks to ensure label quality in the training data construction pipeline, we conduct a small scale human evaluation to measure the quality of the generated data.

We randomly chose 40 (document, claim) pairs from each of the C2D and D2C training data. Three authors of the work independently annotated these 40×2 datapoints as supported or not supported (without seeing the gold label), with an average annotation time of around 1 min and 40 seconds per example. Among the three annotators, we computed an annotation agreement using Fleiss' Kappa, obtaining a score of 0.51 for the C2D samples and 0.70 on the D2C samples, indicating moderate to substantial agreement. The annotators adjudicated cases with disagreement to reach a consensus on the final factuality of the labels. We refer to these as the ground truth labels.

Compared to these ground truth labels, the labels given to our training samples have an accuracy of 80% on C2D and 78% on D2C, respectively. We calculated similar accuracy values for the human annotators, and the average across the three annotators' accuracies was 85% on C2D and 88% on D2C. These accuracies demonstrate that the automatic labels are a bit lower-quality than single-annotator labels on D2C, but close on C2D.

Many of the disagreement cases are classic cases of subjectivity in NLI (Pavlick and Kwiatkowski, 2019; Chen et al., 2020; Nie et al., 2020b). For example, for the C2D claim "*Located in the Scottish Highlands, she lived with and later married James Ballard in Spean Bridge.*", the generated passage only references the couple living in the town of Spean Bridge after the marriage, making it unclear whether or not they got married there. However, this is a reasonable supposition to make. Our results show that our data is useful for training in spite of these subjective examples.

B Additional Results

B.1 Results using Threshold Tuning

As shown in Table 7, AlignScore achieves the highest overall performance (72.2%) on LLM-

Model Name	LLM-AGGREFACT (decomposition - *without* threshold tuning)										
	AGGREFACT		TOFUEVAL		WICE	REVEAL	CLAIM VERIFY	FACT CHECK	EXPERT QA	LFQA	Avg
	CNN	XSum	MediaS	MeetB							
GPT-4	70.1	74.3	70.9	79.4	77.6	87.4	73.2	79.9	59.6	84.9	75.6 (↑ **0.3**)
SummaC-CV	65.2	51.2	58.0	55.2	50.9	66.2	69.6	53.4	53.5	65.0	58.8 (↓ **3.3**)
QAFactEval	61.7	60.4	65.0	60.2	59.1	81.7	68.6	66.1	54.6	73.0	64.6 (↓ **1.9**)
SummaC-ZS	62.4	64.5	64.5	70.0	64.8	85.6	70.0	75.2	56.3	77.2	69.1 (↑ **1.2**)
AlignScore	65.6	68.3	71.2	73.2	63.9	86.2	73.7	74.3	57.6	82.6	71.5 (↑ **1.1**)
MiniCheck-RBTA	65.0	72.1	72.6	75.1	66.9	88.5	76.1	73.3	58.0	84.3	73.2 (↑ **0.5**)
MiniCheck-DBTA	59.9	73.3	67.9	74.5	75.0	88.2	72.5	73.0	57.9	84.5	72.7 (↑ **0.1**)
MiniCheck-FT5	65.9	71.7	69.2	77.4	72.9	87.0	73.5	74.7	57.5	83.2	73.3 (↓ **1.4**)

Table 8: Performance of models on the test set of LLM-AGGREFACT by aggregating predictions on decomposed claims. We include the performance change compared to predicting using original claims from Table 2 (red for worse performance and green for better performance).

Model Name	LLM-AGGREFACT (decontextualization - *without* threshold tuning)										
	AGGREFACT		TOFUEVAL		WICE	REVEAL	CLAIM VERIFY	FACT CHECK	EXPERT QA	LFQA	Avg
	CNN	XSum	MediaS	MeetB							
GPT-4	66.7	76.5	71.5	79.2	80.3	87.0	66.2	79.9	60.1	86.4	75.3 (+**0.0**)
SummaC-CV	65.2	54.5	62.6	62.1	52.5	67.8	69.0	53.4	54.7	66.5	60.8 (↓ **1.3**)
QAFactEval	54.3	62.1	62.8	64.5	62.7	82.8	71.7	66.9	56.5	80.3	66.4 (↓ **0.1**)
SummaC-ZS	51.1	61.5	67.5	71.4	63.0	85.8	69.0	75.2	56.4	76.1	67.7 (↓ **0.2**)
AlignScore	52.4	71.4	68.8	72.2	65.1	85.5	69.1	74.3	59.6	85.2	70.4 (+**0.0**)
MiniCheck-RBTA	63.7	70.8	70.9	75.6	64.3	88.9	76.0	73.3	57.5	83.2	72.4 (↓ **0.3**)
MiniCheck-DBTA	64.2	71.0	69.6	69.1	63.3	87.5	73.6	73.0	57.8	83.2	71.2 (↓ **1.4**)
MiniCheck-FT5	69.9	74.3	74.0	75.6	69.7	86.2	73.0	74.7	58.4	85.2	74.1 (↓ **0.6**)

Table 9: Performance of models on the test set of LLM-AGGREFACT by doing claim decontextualization where it is applicable. We include the performance change compared to predicting using original claims from Table 2.

AGGREFACT among fact-checkers from prior work. However, by fine-tuning AlignScore's backbone RoBERTa model on our 14K synthetic data (MiniCheck-RBTA), we surpass AlignScore's performance by 1.5%. This improvement is more significant in the setting without threshold tuning (Section 6). Remarkably, this boost in performance is attained using a dataset that constitutes less than 0.3% of the total data on which AlignScore was initially trained (Table 3). This finding highlights the potential of curated synthetic data in enhancing the performance of state-of-the-art fact-checkers. Overall, our models achieve new state-of-the-art among specialized models under the threshold tuning setting.

Our synthetic data enhances model robustness and performance in the absence of threshold tuning. Comparing Table 2 with Table 7, we see that the performance of specialized fact-checkers decreases without threshold tuning. However, *MiniCheck-FT5 only drops by 0.4%* compared to larger drops for other systems, such as 9.2% for SummaC-CV. These results suggest that our synthetic data not only improves overall performance but also enhances the robustness of models across the domains of our benchmark, enabling them to maintain strong performance even without threshold tuning.

B.2 Full Results Using Claim Decomposition

In Table 8, we show the performance (and performance changes) of GPT-4 and a subset of specialized fact-checkers across all datasets from LLM-AGGREFACT, using claim decomposition to determine the factuality label for each claim.

B.3 Full Results Using Claim Decontextualization

In Table 9, we show the performance (and performance changes) of GPT-4 and a subset of specialized fact-checkers across all datasets from LLM-AGGREFACT, using claim decontextualization when applicable. In particular, we perform claim decontextualization on TOFUEVAL-MediaS,

Model Name	TofuEval	
	MediaS	MeetB
GPT-4	71.4	79.9
GPT-4-Full	72.3	79.7

Table 10: Comparison of GPT-4 and GPT-4-Full on TofuEval, a dataset where sentences from an LLM-generated response share the same grounding document.

TofuEval-MeetB, Wice, Reveal, ClaimVerify, ExpertQA, and Lfqa. Note that the claim decontextualization step add a non-negligible amount of cost, as shown in Table 4.

B.4 Results Predicting All Errors

When a grounding document is relevant to multiple sentences in a response, it becomes feasible to ask an LLM-based fact-checker to simultaneously predict the factuality labels for all sentences, thereby reducing cost. We investigate this approach using GPT-4 on the TofuEval datasets, where we provide GPT-4 with a document and the entire summary, asking the model to predict the factuality labels for all summary sentences at once, denoted as **GPT4-Full**. The prompt is shown in Table 28. Table 10 shows that GPT4-Full achieves performance similar to predicting the factual label for each summary sentence individually. The inference cost on the TofuEval test set can be reduced from $16.7 to $6.72 with this method.

However, in retrieve-then-generate and post-hoc grounding settings, evidence is typically retrieved for each claim separately, meaning no document can be shared across claims in a single response, and thus the inference cost is barely reduced.

C LLM-AggreFact Details

C.1 Dataset Descriptions

AggreFact (Tang et al., 2023a) is a factual consistency evaluation benchmark for new summarization, targeting **CNN(/DM)** (Nallapati et al., 2016) and **XSum** (Narayan et al., 2018). Our focus is on the SOTA sets within AggreFact, where summaries are generated from SOTA fine-tuned summarizers, since their analysis suggests that summaries are more challenging to evaluate for factual consistency compared to summaries generated by pre-SOTA summarizers. Data in AggreFact comes from 9 factual consistency evaluation datasets on CNN or XSum, including widely used ones such as SummaC (Laban et al., 2022), FRANK (Pagnoni et al., 2021), and SummEval (Fabbri et al., 2021). Check Appendix C for a complete set of evaluation datasets in AggreFact. Because CNN/DM and XSum feature quite different styles of summaries, we report these numbers separately in our benchmark. However, we do not otherwise report results on the smaller datasets within the AggreFact SOTA subset.

TofuEval (Tang et al., 2024) is a factual consistency evaluation benchmark for dialogue summarization, targeting **MediaS**um (interviews, Zhu et al. (2021)) and **Meet**ing**B**ank (city council meetings, Hu et al. (2023)). It includes topic-focused dialogue summaries generated by 6 LLMs, with sentence-level factual consistency annotations by linguists.

Wice (Kamoi et al., 2023) is a textual entailment dataset that consists of naturally occurring claims from Wikipedia and their cited documents. Based on its cited documents, each claim is labeled as `supported`, `partially-supported`, or `non-supported`.

Reveal (Jacovi et al., 2024) is a benchmark dataset that evaluates the correctness of reasoning chains generated by LLMs in the context of open-domain question-answering. The dataset includes annotations at the sentence level, covering various aspects of response correctness. For our dataset, we focus on the subset of sentences that have *attribution* annotations, which indicate whether a sentence in a reasoning chain can be attributed to information retrieved from Wikipedia paragraphs with three label categories: `fully attributable`, `partially attributable`, or `contradictory`.

ClaimVerify (Liu et al., 2023) evaluates the correctness of responses from four generative search engines in answering user queries. Similar to Wice, the dataset contains annotations on whether check-worthy sentences from the engines' responses can be fully supported by their associated cited documents. The dataset contains binary-level factual consistency annotations for each cite-worthy sentence.

FactCheck-GPT (Wang et al., 2023) contains factual consistency annotations for LLMs' responses to search queries. In this dataset, each sentence from LLMs' responses is first decomposed into atomic facts and those atomic facts are then fur-

ther decontextualized so that they can stand alone. Finally, each worth-checking and decontextualized atomic fact is labeled as `completely support`, `partially support`, `refute`, or `irrelevant`. We include those decontextualized atomic facts and their corresponding documents in the benchmark.

EXPERTQA (Malaviya et al., 2024) contains responses from 6 different systems to queries curated by experts from 32 fields. These systems answer queries either in a close-book fashion with/without in-line citations, or based on retrieved document(s). For each sentence in the response, the sentence is verified against the concatenation of cited or retrieved document(s), if any. We include examples where documents are judged as `complete`, `partial`, or `incomplete` in supporting the corresponding sentences. We do not include human edited claims and evidence in our benchmark.

LFQA (Chen et al., 2023) contains LLM-generated responses to questions from the ELI5 ("Explain Like I'm Five") dataset (Fan et al., 2019). LLMs generate responses based on documents that are either retrieved by humans, models, or randomly selected. Human annotators then evaluate each sentence in the LLM-generated responses against the corresponding document set, classifying them into `supported`, `partially supported`, or `not supported`.

C.2 Label Unification

For AGGREFACT, TOFUEVAL, and CLAIMVERIFY, we keep using the binary label from the original work. For the remaining datasets, we map `supported`, `fully attributable`, `completely support`, and `complete` to `supported`, and `unsupported` otherwise.

C.3 Excluded Datasets

We excluded **HALUEVAL** (Li et al., 2023) and **SUMMEDITS** (Laban et al., 2023) from our benchmark since they are synthetic, with errors in summaries generated via instruction prompts that guide the model to intentionally make errors in summaries. These errors are unnatural and do not fit with our goal of detecting true LLM generation errors. **FACTSCORE** (Min et al., 2023) contains naturally generated biographies from LLMs and has human-annotated labels of individual atomic facts. However, humans could potentially search different articles to verify the correctness of those sentences than the ones that models retrieve. As a result, a non-negligible fraction of the claims in the dataset appear mislabeled from the standpoint of the fact-checking on grounded documents task.

Dataset		Split	Size	Doc Len	Claim Len	% of Neg
AGGREFACT	CNN	dev	459	558	56	13%
		test	558	563	58	10%
	XSum	dev	777	374	26	49%
		test	558	370	25	49%
TOFUEVAL	Media	dev	1800	990	21	20%
		test	726	922	21	24%
	MeetB	dev	1607	930	22	18%
		test	772	915	22	19%
WICE		dev	349	1622	27	67%
		test	358	1683	28	69%
REVEAL		dev	1656	104	11	77%
		test	1710	103	11	77%
CLAIM VERIFY		dev	1093	1874	21	25%
		test	1088	1841	22	27%
FACT CHECK		dev	1537	100	14	83%
		test	1566	100	13	76%
EXPERT QA		dev	3773	491	29	22%
		test	3702	506	29	20%
LFQA		dev	2029	383	25	43%
		test	1911	380	25	41%

Table 11: **Statistics of datasets in LLM-AGGREFACT.** We show the size of datasets, the average length of documents and claims, and the proportion of unsupported claims.

C.4 Statistics

The statistics of LLM-AGGREFACT can be found in Table 11. Our use of these datasets is for research purposes only, which is consistent with their intended use.

AGGREFACT contains the following 9 factual consistency evaluation datasets on CNN or XSum: FactCC (Kryscinski et al., 2020), Wang'20 (Wang et al., 2020), SummEval (Fabbri et al., 2021), Polytope (Huang et al., 2020), Cao'22 (Cao et al., 2022), XSumFaith (Maynez et al., 2020), FRANK (Pagnoni et al., 2021), Goyal'21 (Goyal and Durrett, 2021), and CLIFF (Cao and Wang, 2021).

D Synthetic Data Details

Source of Data In our C2D method, we choose around 400 claims from Wikipedia that have cited web articles (Kamoi et al., 2023; Petroni et al., 2023) to generate synthetic documents. In our D2C method, we scraped around 300 Google News articles since November 2023 from diverse topic cate-

	Label	Min.	25%	50%	75%	Max.
C2D	1	72	147	182	242	359
	0	66	136	171	234	359
D2C	1	85	139	162	186	427
	0	84	138	161	186	493

Table 12: Length distribution of generated documents. We use the NLTK package for word tokenizatiopn.

gories, including science, politics, world, entertainment, business, and technology. Each document is approximately 500 words. Statistics of our generated data can be found in Table 1 and 12. See Appendix H.1 for how we maintain the quality of our synthetic data.

Characteristics of Synthetic Data It is worth noting that constructing our synthetic dataset involves using human-written or naturally generated claims, which sets it apart from prior synthetic data generation methods used to train fact-checkers for text summarization. These methods, such as entity swapping and sentence negation (Kryscinski et al., 2020; Goyal and Durrett, 2021), were designed to target specific error types that occurred in claims from earlier summarization models. However, as errors from generative models progress (Tang et al., 2023a) and new error types emerge from LLMs, focusing on specific error types may not generalize well to unseen datasets with potentially novel errors.

Examples of synthetic data for C2D and D2C can be found in Table 15 and 16.

Data Rejection Rate Since we use GPT-4 for filtering out low-quality examples in our C2D method, we report the rejection rate at different steps that require entailment checks.

During the *atomic fact expansion* step (step 2), 6% of the final generated sentence pairs, when combined, could not support the original atomic fact. In the *supporting document generation* step (step 3), 5% of the final documents failed the entailment check. For the *non-supporting document generation* step (step 4), 53% of the final documents still supported the claim, and these documents were not included in our constructed data. These filtering steps are crucial for improving the training dataset's quality.

Model	Checkpoint
Gemini-Pro	`gemini-1.0-pro`
PaLM2-Bison	`chat-bison@001`
Mistral-8x7B	`open-mixtral-8x7b`
Mistral-Large	`mistral-large-2402`
Claude-2.1	`claude-2.1`
Claude-3 Opus	`claude-3-opus-20240229`
GPT-3.5	`gpt-3.5-turbo-0125`
GPT-4	`gpt-4-0125-preview`

Table 13: LLM checkpoints

E Fact-Checking Model Details

E.1 LLM-Based Fact-Checkers

We use the official APIs for LLM-based fact-checkers. The checkpoints we use for LLMs can be found in Table 13. The inference prompt is the same for all LLMs and can be found in Table 23. We use a temperature of zero to collect deterministic outputs, which is typical from previous work.

E.2 Specialized Fact-Checkers

QAFactEval (Fabbri et al., 2022) is a QA-based fact-checker with optimized components for answer selection, question answering, question generation, and answer overlap calculation. It selects spans as answers from a summary sentence, generates questions based on these answers, and then answers these questions using the source document. Finally, it computes an overall overlap score for the summary sentence by comparing the selected spans from the summary sentence with the answers derived from the source document, given the generated questions. QAFactEval produces scores on a continuous scale ranging from 0 to 5. In our experiments, we use the default model and hyperparameters as provided by the authors.

DAE (Goyal and Durrett, 2020, 2021) is an entailment-based fact-checker that evaluates the factual consistency of each dependency arc in a summary sentence. It independently verifies whether the semantic relationship of each dependency arc is factually supported by the source document. Finally, it aggregates the scores for all dependency arcs to compute an overall sentence-level factuality score ranging from 0 to 1. In our experiments, we use the default model and hyperparameters as provided by the authors in Goyal and Durrett (2021).

SummaC-ZS (Laban et al., 2022) is an entailment-based fact-checker. To evaluate a summary sentence c_i, it divides the source document D_i into a set of sentences or paragraphs $D_i = \{d_{i,1}, \ldots, d_{i,|d|}\}$, and the score for c_i is determined by the highest score among all $(d_{i,j}, c_i)$ pairs, i.e., $\text{score}(c_i) = \max_j M(di, j, c_i)$. For a multi-sentence summary, the final score is calculated as the average of the individual sentence scores. In our experiments, we do not use the authors' default setting of splitting the document D_i into sentences and instead choose paragraph-level segmentation, as most datapoints in LLM-AGGREFACT require reasoning across multiple sentences. We find this change not only improves the overall performance but also accelerates inference speed. Apart from this adjustment, we adhere to the default model and hyperparameters provided by the authors. SummaC-ZS returns a score between -1 and 1.

SummaC-CV (Laban et al., 2022) extends SummaC-ZS by considering all entailment scores for each summary sentence c_i. Similar to SummaC-ZS, SummaC-Conv evaluates a summary sentence c_i by dividing the source document D_i into $D_i = \{d_{i,1}, \ldots, d_{i,|d|}\}$. However, instead of selecting the maximum score among all $(d_{i,j}, c_i)$ pairs, SummaC-Conv uses a learned convolutional layer to transform the distribution of entailment scores $\{M(d_{i,j}, c_i) : \forall j\}$ into a single score. The final summary score is computed by averaging the scores of individual sentences. As with SummaC-ZS, we use paragraph-level segmentation in our experiments and keep other settings as default. SummaC-Conv outputs a score between 0 and 1.

AlignScore (Zha et al., 2023) is an entailment-based model that has been trained on data from a wide range of tasks such as NLI, QA, fact verification, and summarization. It works similarly to SummaC-ZS, with the only difference being that it splits a document $D_i = \{d_{i,1}, \ldots, d_{i,|d|}\}$ into sequential chunks at sentence boundaries. Each chunk contains approximately 350 tokens, determined by white space splitting. In our experiments, we use the default model and hyperparameters as provided by the authors. AlignScore outputs a score between 0 and 1.

T5-NLI-Mixed (Honovich et al., 2022) is an entailment-based fact-checker built on T5-XXL. It has been trained on a diverse set of NLI datasets and predicts whether a given claim is supported by a document, outputting "1" for supported claims and "0" for unsupported ones. The final entailment score is calculated as the probability of the model predicting the token "1". To optimize its performance on 2 GPUs from our hardware setup, we select a chunk size of 350 tokens according to the T5 tokenizer. T5-NLI-Mixed outputs a score between 0 and 1.

MINICHECK-RBTA, MINICHECK-DBTA also split a document into chunks at sentence boundaries, with a chunk size of approximately 400 tokens according to RoBERTA and DeBERTa tokenizers. This results in approximately the same chunk size as in AlignScore, which has a chunk size of 350 tokens using white space splitting. The output scores fall within the range of 0 to 1.

FT5-ANLI-L, MiniCheck-FT5 work the same way as T5-NLI-Mixed, but using only one GPU and setting the chunk size to 500 tokens using white space splitting. The output scores fall within the range of 0 to 1.

E.2.1 Machine Configuration for Specialized Fact-Chekers

We use two `NVIDIA RTX A6000` GPUs for T5-NLI-Mixed, given its model size, and one GPU for the remaining models, all on our own hardware. According to Lambda,[6] a single `NVIDIA RTX A6000` GPU costs $0.8 per hour and has 48 GB VRAM.

F Baseline Synthetic Data Generation Methods

We describe the simplified methods in generating C2D and D2C datasets, denoted as C2D-SIMP and D2C-Simp. Performance on models trained on those simplified synthetic datasets can be found in Section A.1.

The motivation for these models is to capture the performance of a more basic prompting approach, where we simply ask GPT-4 to generate a data instance in one shot. Comparing the performance of this with C2D/D2C helps validate our more sophisticated prompting strategy.

C2D-SIMP To generate the C2D-SIMP dataset, we begin by providing GPT-4 with a claim c and asking it to create a supporting document D that requires multiple sentences together to support the

[6] Detailed price specifications are available at https://lambdalabs.com/service/gpu-cloud##pricing.

claim (see Table 25 for the generation prompt). We then ask GPT-4 to minimally modify D to create a new document D', which can support some atomic facts mentioned in c but not all of them. Inspired by the error type definitions from Tang et al. (2023a), we provide four different revision types to help GPT-4 generate diverse non-supporting documents, covering various reasons for not supporting the claim (see Table 26 for the prompt). As the generated supporting documents for a given claim tend to be similar despite adjusting the model temperature, we do not generate multiple supporting documents for c. Instead, for each claim, we generate one supporting and pair it with one non-supporting document. To enhance the diversity of the training data and maintain a comparable dataset size to our C2D method, we randomly select 3,500 claims from Wikipedia with cited web articles, resulting in the C2D-SIMP dataset containing 7K datapoints.

D2C-SIMP We start by directly using the summary sentences generated using the chunk-level summarization step of our D2C method (Section 3.2). That is, for each human written document, we have three document chunks $\{D_1, D_2, D_3\}$ and corresponding supporting summary sentences $\{c_1, c_2, c_3\}$ generated by GPT-4. For each $(D_i, c_i, 1)$ tuple, we ask GPT-4 to modify the summary sentence c_i such that the edited summary sentence c_i' is no longer supported by the document chunk D_i. We leverage the editing method from Laban et al. (2023), which is used to construct their SummEdit factual consistency evaluation benchmark. The editing prompt is provided in Table 27. We sample 7K datapoints from the generated data to construct D2C-SIMP.

G Training Details

We include the training details and hyperparemater details in the section. Unless otherwise specified, we use the default hyperparematers of the backbone models. All models are trained using the standard cross-entropy loss function.

For our baseline models: FT5-C2D, FT5-D2C, FT5-ANLI-L, FT5-C2D-S, and FT5-D2C-S, we fine-tune `flan-t5-large`[7] for 2 epochs on prepared data described in Section 5 and A, using a batch size of 4 and a learning rate of 5e-5.

For MiniCheck-RBTA, MiniCheck-DBTA and MiniCheck-FT5, we begin by fine-tuning the tuned RoBERTa-Large model from AlignScore, `deberta-v3-large`[8], and `flan-t5-large` on their respective training data (Section 3.3) for 2 epochs, while excluding 7K D2C synthetic data. We use a batch size of 4 with an accumulation step of 2 and a learning rate of 1e-5 for RoBERTa and 5e-5 for the other two models. We then fine-tune these models on 7K D2C synthetic data for 1 epoch, with a batch size of 4 and learning rate of 1e-5.

We observe that following this training pipeline consistently yields higher performance across all three backbone models compared to training on all data simultaneously. We hypothesize that this improvement stems from the fact that the source documents in the D2C dataset are human-written documents, in contrast to the synthetically generated source documents in the C2D dataset. Fine-tuning on these realistic documents at the end helps the models adapt back to a realistic distribution, preventing them from overfitting to synthetic documents and allowing them to perform well on real documents in the benchmark.

Prompt Functionality	**Ref.**
Sentence decomposition	Table 17
Atomic fact expansion (C2D)	Table 18
Document generation (C2D)	Table 19
Supporting doc. generation (C2D-SIMP)	Table 25
Non-supporting doc. generation (C2D-SIMP)	Table 26
Merging atomic facts (C2D, D2C)	Table 22
Chunk-level summarization (D2C, D2C-Simp)	Table 21
Non-supporting doc. generation (D2C-SIMP)	Table 27
Entailment check for data construction	Table 20
Zero-shot factual consistency evaluation	Tabel 23
Sentence decontextualization	Table 24

Table 14: References to prompts. Upper: prompts for our synthetic data generation methods, and simplified synthetic data generation methods. Lower: Prompts for evaluation on LLM-AGGREFACT.

H Prompts

In Table 14, we present the full list of the prompts used throughout our work. We use GPT-3.5 for sentence decomposition and merging atomic facts, and GPT-4 for the remaining prompts. We next elaborate on how we ensure the labeling quality of our synthetically generated data.

H.1 Quality Assurance for Generations

Sentence decomposition We adapt a few-shot sentence decomposition prompt from (Kamoi et al.,

[7] huggingface.co/google/flan-t5-large

[8] huggingface.co/microsoft/deberta-v3-large

2023), which can generate complete and correct atomic facts most of the time according to their human evaluation. The prompt (Table 17) is used for both of our synthetic data generation methods and the claim decomposition experiment in Section 7.1.

C2D: Atomic fact expansion We use a 4-shot prompt (Table 18) for this step, where we ask GPT-4 to produce a sentence pair where the atomic fact is supported if and only if the information from both sentences is combined. To ensure the quality of the generation, we verify the correctness of this condition after generation via an entailment check by GPT-4 (Table 20). If the correctness is not met, we iterate the process and regenerate a new sentence pair up to a specified number of attempts. In cases where the correctness criterion remains unmet after the specified attempts, we remove the datapoint from the dataset.

C2D: Supporting document generation We ensure that all sentences $\mathbf{s}$ from the generated sentence pairs are mentioned in (and hence are entailed by) the generated document D by using the entailment check by GPT-4. Same as above, if the document fails to mention all sentences from the sentence pairs, we iteratively generate new documents until a specified number of attempts is reached. It is important to note that we only verify whether sentences $\mathbf{s}$ are mentioned in D, as we believe GPT-4 can perform well on this simple task. By construction, if all sentences are mentioned in D, then c is supported by D. However, directly performing an entailment check on the (D, c) pair with GPT-4 may introduce many labeling errors, which can negatively impact the performance of the trained models.

Document	Claim	Label
More than 5,000 individuals, part of a caravan that crossed into Mexico last month, are now seeking asylum and have established a temporary encampment at the Tijuana Stadium as of today. The Tijuana Stadium, known for hosting sporting events, recently underwent renovations that doubled its seating capacity. Prior to these changes, the stadium had a capacity to accommodate 1,500 spectators.	By this date, over 5,000 members of the caravan were staying at the Tijuana Stadium — a structure with a capacity of 3,000.	S
	By this date, over 5,000 members of the caravan were staying at the Tijuana Stadium.	S
	The Tijuana Stadium has a capacity of 3,000.	S
More than 5,000 individuals who are part of a caravan that crossed into Mexico last month have now established a temporary encampment at the Tijuana Stadium, where they are reportedly seeking asylum. The stadium, known for hosting sporting events, could originally accommodate 1,500 spectators before it became the site of the makeshift settlement. As of today, the facility is being repurposed to provide the asylum seekers with temporary shelter as they await the processing of their claims.	By this date, over 5,000 members of the caravan were staying at the Tijuana Stadium — a structure with a capacity of 3,000.	U
	By this date, over 5,000 members of the caravan were staying at the Tijuana Stadium.	S
	The Tijuana Stadium has a capacity of 3,000.	U
As of today, a group of individuals has established a temporary encampment within the premises of the Tijuana Stadium, according to officials. The stadium, which has recently undergone extensive renovations that included an expansion to double its original capacity, can now welcome a significantly larger audience. Prior to the upgrade, the Tijuana Stadium was known to have a seating capacity for 1,500 spectators; the recent improvements are expected to enhance its utility for various events and gatherings.	By this date, over 5,000 members of the caravan were staying at the Tijuana Stadium — a structure with a capacity of 3,000.	U
	By this date, over 5,000 members of the caravan were staying at the Tijuana Stadium.	U
	The Tijuana Stadium has a capacity of 3,000.	S
In a significant movement at the border, a caravan comprising over 5,000 asylum seekers penetrated Mexico's boundaries last month, bringing to the forefront the ongoing challenges faced by migrants from multiple origins. The group has today established a makeshift camp within the confines of the Tijuana Stadium, a venue known for its recent renovation that doubled its seating capacity. The temporal shift marks a new chapter for the individuals on their quest for safety and stability, with the stadium offering a transient sanctuary as they navigate their next steps.	By this date, over 5,000 members of the caravan were staying at the Tijuana Stadium — a structure with a capacity of 3,000.	U
	By this date, over 5,000 members of the caravan were staying at the Tijuana Stadium.	S
	The Tijuana Stadium has a capacity of 3,000.	U

Table 15: **Examples using the C2D method.** Documents are generated from the claim (from Wikipedia): *By this date, over 5,000 members of the caravan were staying at the Tijuana Stadium — a structure with a capacity of 3,000.* The same claim can be both supported (S) and unsupported (U) by documents, which encourage models to pay attention to multiple atomic facts in a sentence. Determining the factuality labels of claims requires models to reason over multiple sentences. Unsupporting facts are highlighted.

Document	Claim	Label
(Doc Chunk 1) With the SAG-AFTRA strike settled, the six-month, multi-guild Hollywood labor disruption has finally ended, but the theatrical damage has only begun to surface. Reviewing the films delayed until next year, a rough estimate suggests that the stoppage cost theaters around $400 million- $600 million in gross – more, when including lost concession revenue. "Barbie," "Oppenheimer," "Sound of Freedom," and "Taylor Swift: The Eras Tours" kept the damage from being worse. Immediately following the SAG-AFTRA settlement, Disney announced wholesale delays in its upcoming release schedule. Their revived plans includes only one Marvel title for 2024 ("Deadpool 3," moved to July 26 from May 3), down from the customary three per year from MCU. Disney was among the first studios to announce delays, with Sony already out Wednesday evening with word the third "Venom" film would move from July to November. Related Stories The good news for theaters is despite it all, 2023 should still reach the $9 billion in domestic gross hoped for this year.	2024 box-office hopes dashed by production delays and major title postponements, costing potentially $500 million.	S
(Doc Chunk 2; corresponding chunk) However, any hopes that 2024 might return to 2019 box-office parity are dashed. Grosses from rescheduled titles will help, but production delays will leave substantial gaps. Even so: It could have been worse. "Dune: Part 2" (Warner Bros.) and "Kraven the Hunter" and "Ghostbusters: Frozen Empire" (Sony) will cost this year's total the most – perhaps $400 million ("Ghostbusters" would have had only 12 days of 2023 play). Figure other films, mostly limited/specialized entries like Luca Guadagnino's "Challengers" and Jeff Nichols "The Bikeriders" (Disney), Ethan Coen's "Drive Away Dolls" (Focus), and "The Book of Clarence" (Sony) could have contributed $100 million or more while riding the awards wave. The biggest unknown is how much the lack of promotion hurt the films released during the strikes. By the time the SAG-AFTRA strike began July 11, most of the summer's top titles had already been released or were about to be, which meant their promotional pushes were all but complete.	2024 box-office hopes dashed by production delays and major title postponements, costing potentially $500 million.	S
(Doc Chunk 2; corresponding chunk) ~~However, any hopes that 2024 might return to 2019 box-office parity are dashed.~~ Grosses from rescheduled titles will help, but production delays will leave substantial gaps. Even so: It could have been worse. "Dune: Part 2" (Warner Bros.) and "Kraven the Hunter" and "Ghostbusters: Frozen Empire" (Sony) will cost this year's total the most – perhaps $400 million ("Ghostbusters" would have had only 12 days of 2023 play). Figure other films, mostly limited/specialized entries like Luca Guadagnino's "Challengers" and Jeff Nichols "The Bikeriders" (Disney), Ethan Coen's "Drive Away Dolls" (Focus), and "The Book of Clarence" (Sony) could have contributed $100 million or more while riding the awards wave. The biggest unknown is how much the lack of promotion hurt the films released during the strikes. By the time the SAG-AFTRA strike began July 11, most of the summer's top titles had already been released or were about to be, which meant their promotional pushes were all but complete.	2024 box-office hopes dashed by production delays and major title postponements, costing potentially $500 million.	U
(Doc Chunk 2; corresponding chunk) However, any hopes that 2024 might return to 2019 box-office parity are dashed. Grosses from rescheduled titles will help, but production delays will leave substantial gaps. Even so: It could have been worse. "Dune: Part 2" (Warner Bros.) and "Kraven the Hunter" and "Ghostbusters: Frozen Empire" (Sony) will cost this year's total the most – perhaps $400 million ("Ghostbusters" would have had only 12 days of 2023 play). ~~Figure other films, mostly limited/specialized entries like Luca Guadagnino's "Challengers" and Jeff Nichols "The Bikeriders" (Disney), Ethan Coen's "Drive Away Dolls" (Focus), and "The Book of Clarence" (Sony) could have contributed $100 million or more while riding the awards wave.~~ The biggest unknown is how much the lack of promotion hurt the films released during the strikes. By the time the SAG-AFTRA strike began July 11, most of the summer's top titles had already been released or were about to be, which meant their promotional pushes were all but complete.	2024 box-office hopes dashed by production delays and major title postponements, costing potentially $500 million.	U
(Doc Chunk 3) Some, like DC Comics' "Blue Beetle" (WB) and "The Equalizer 3" (Sony), may have suffered more. Would promotion for all strike-period releases have total $100 million? Maybe. Theaters were fortunate that both "Barbie" and "Oppenheimer" already had enormous publicity before actors struck, and both had major, Oscar-nominated directors to carry the ball. Their $950 million combined domestic gross more than doubled expectations. Add the mid-summer sleeper success of "Sound of Freedom" and July and August both were strong months. A wild card, unknown when the strike began, was Taylor Swift's concert film. It certainly filled an October void that existed before the strike and its SAG-AFTRA waver meant she could promote it. The outside-studio success of "Sound of Freedom" and "The Eras Tour" were not only welcome for their grosses, but also because they show it's possible to find releases outside the studios.	2024 box-office hopes dashed by production delays and major title postponements, costing potentially $500 million.	U

Table 16: **Examples using the D2C method.** The full document is from this website. The same claim (summary sentence) can be supported by both its directly associated document chunk (chunk 2) and a separate chunk (chunk 1) originating from the same document, which has been divided into three distinct chunks. Some ~~sentences~~ are removed from the chunk to make the claim unsupported. Unsupporting facts are highlighted.

Segment the following sentence into individual facts:

Sentence: Other title changes included Lord Steven Regal and The Nasty Boys winning the World Television Championship and the World Tag Team Championship respectively.
Facts:
- Lord Steven Regal won the World Television Championship.
- The Nasty Boys won the World Tag Team Championship.

Sentence: The parkway was opened in 2001 after just under a year of construction and almost two decades of community requests.
Facts:
- The parkway was opened in 2001.
- The parkway was opened after just under a year of construction.
- The parkway was opened after two decades of community requests.

Sentence: Touring began in Europe in April-June with guitarist Paul Gilbert as the opening act, followed by Australia and New Zealand in July, Mexico and South America in late July-August, and concluding in North America in October-November.
Facts:
- Touring began in Europe in April-June.
- The opening act of the tour was guitarist Paul Gilbert.
- The tour was in Australia and New Zealand in July.
- The tour was in Mexico and South America in late July-August.
- The tour was concluded in North America in October-November.

Sentence: In March 2018, the company partnered With Amazon Web Services (AWS) to offer Al-enabled conversational solutions to customers in India.
Facts:
- The company partnered with Amazon Web Services (AWS) in March 2018.
- The two companies partnered to offer Al-enabled conversational solutions to customers in India.

Sentence: The most significant of these is in Germany, which now has a Yazidi community of more than 200,000 living primarily in Hannover, Bielefeld, Celle, Bremen, Bad Oeynhausen, Pforzheim and Oldenburg.
Facts:
- The most significant of these is in Germany.
- Germany now has a Yazidi community of more than 200,000.
- Yazidi community in Germany lives primarily in Hannover, Bielefeld, Celle, Bremen, Bad Oeynhausen, Pforzheim and Oldenburg.

Sentence: A previous six-time winner of the Nations' Cup, Sebastian Vettel became Champion of Champions for the first time, defeating Tom Kristensen, who made the final for the fourth time, 2-0.
Facts:
- Sebastian Vettel is a previous six-time winner of the Nations' Cup.
- Sebastian Vettel became Champion of Champions for the first time, defeating Tom Kristensen, 2-0.
- Tom Kristensen made the final for the fourth time.

Sentence: [SENTENCE]
Facts:

Table 17: Sentence decomposition prompt adapted from (Kamoi et al., 2023).

Your task is to generate a pair of sentences so that the provided claim can be entailed by the sentence pair. You must make sure that the claim can only be deduced by combining the information from the two sentences that contain unique information.

Examples:
Provided Claim: The investigation is into allegations that his mayoral campaign received illegal foreign funds.
Sentence 1: During the period leading up to the mayoral election, there was a notable increase in his campaign's financial resources.
Sentence 2: Investigation shows the funds having origins beyond national boundaries, a detail raising questions under current campaign laws.

Provided Claim: Approximately 1,000 fans fainted at the concert.
Sentence 1: Emergency services reported an unusually high number of calls for medical assistance during the concert with an attendance of 20,000.
Sentence 2: Venue officials estimated that approximately 5% of the audience required medical attention for fainting.

Provided Claim: The interest rate hikes were intended to manage inflation and moderate economic growth.
Sentence 1: Central bank officials expressed concern over the rising consumer price index and the overheating of the economy.
Sentence 2: The monetary policy committee decided to adjust the interest rates as a response to these economic indicators.

Provided Claim: Several advertisers are considering halting their ads on social media platform X.
Sentence 1: Some companies are re-evaluating their marketing strategies to avoid association with platforms that fail to address misinformation.
Sentence 2: Recent reports show that platform X has received criticism for its handling of false information spreading unchecked.

Please make sure that NEITHER sentence alone supports the claim.

Your turn:
Provided Claim: [CLAIM]

Table 18: Prompt for *Step 2: Atomic fact expansion* for the C2D method (Section 3.1).

We are creating a news article (one paragraph) in the style of The New York Times. We will give you a list of facts to use when writing your article. You must include all the facts in the list. Never state deduced facts or conclusions. The article should stick to the fact list pretty closely. Include as many sentences as needed to write each fact from the list of facts.

Facts you must include:
-{FACT1}
-{FACT2}
...

Table 19: Prompt for *Step 3: Document generation* for the C2D method (Section 3.1).

Source: [SOURCE]
Claim: [CLAIM]

Is the claim fully entailed, or implied, by the source? Please answer with "yes" or "no".

Table 20: Prompt for all steps in our methods that require entailment check.

Document:
[DOCUMENT]

Please generate a summary for the document with the following requirements:
1. The summary should be a fluent and grammatical sentence.
2. The summary should be no more than 15 words.
3. The summary should cover information across the document.
Summary:

Table 21: Summarization prompt for *Step 1: Chunk-level summarization* for the D2C method (Section 3.2) and D2C-SIMP method.

Merge the following individual facts into a single sentence:

Facts:
- Lord Steven Regal wan the World Television Championship.
- The Nasty Boys wan and the World Tag Team Championship.
Sentence: Other title changes included Lord Steven Regal and The Nasty Boys winning the World Television Championship and the World Tag Team Championship respectively.

Facts:
- The parkway was opened in 2001.
- The parkway was opened after just under a year of construction.
- The parkway was opened after two decades of community requests.
Sentence: The parkway was opened in 2001 after just under a year of construction and almost two decades of community requests.

Facts:
- Touring began in Europe in April-June.
- The opening act was guitarist Paul Gilbert.
- There was a tour in Australia in July.
- There was a tour in New Zealand in July.
- There was a tour in Mexico in late July-August.
- There was a tour in South America in late July-August
- The tour was concluded in North America in October-November.
Sentence: Touring began in Europe in April-June with guitarist Paul Gilbert as the opening act, followed by Australia and New Zealand in July, Mexico and South America in late July-August, and concluding in North America in October-November.

Facts:
- The company partnered with Amazon Web Services (AWS) in March 2018.
- The two companies partnered to offer Al-enabled conversational solutions to customers in India.
Sentence: In March 2018, the company partnered With Amazon Web Services (AWS) to offer Al-enabled conversational solutions to customers in India.

Facts:
- The most significant of these is in Germany.
- Germany now has a Yazidi community of more than 200,000.
- Yazidi community in Germany lives primarily in Hannover.
- Yazidi community in Germany lives primarily in Bielefeld.
- Yazidi community in Germany lives primarily in Celle.
- Yazidi community in Germany lives primarily in Bremen.
- Yazidi community in Germany lives primarily in Bad Oeynhausen.
- Yazidi community in Germany lives primarily in Pforzheim.
- Yazidi community in Germany lives primarily in Oldenburg.
Sentence: The most significant of these is in Germany, which now has a Yazidi community of more than 200,000 living primarily in Hannover, Bielefeld, Celle, Bremen, Bad Oeynhausen, Pforzheim and Oldenburg.

Facts:
- Sebastian Vettel is a previous six-time winner of the Nations' Cup.
- Sebastian Vettel became Champion of Champions for the first time.
- Sebastian Vettel defeated Tom Kristensen.
- Tom Kristensen made the final for the fourth time.
- The score was 2-0.
Sentence: A previous six-time winner of the Nations' Cup, Sebastian Vettel became Champion of Champions for the first time, defeating Tom Kristensen, who made the final for the fourth time, 2-0.

Facts:
-{FACT1}
-{FACT2}
...
Sentence:

Table 22: Merging prompt for *Step 5: Pairing subclaims and generated documents* for the C2D method (Section 3.1) and *Step 2: Claim decomposition and subclaim augmentation* for the D2C method (Section 3.2).

Determine whether the provided claim is consistent with the corresponding document. Consistency in this context implies that all information presented in the claim is substantiated by the document. If not, it should be considered inconsistent.
Document: [DOCUMENT]
Claim: [CLAIM]
Please assess the claim's consistency with the document by responding with either "yes" or "no".
Answer:

Table 23: Zero-shot factual consistency evaluation prompt for all LLMs.

You are provied with a context and a claim. Please first determine if the claim can stand alone whitout the conext. If not, provide a decontextualzied version of the claim that incorporates necessary information from the context to make it self-contained. The revision should be as minimum as possible. Please respond with a JSON format: {"label": "yes"/"no", "decontext": "NA"/decontextualized claim}.

Example 1:
Context: There are many reasons why poetry is important for children. Poetry can help children build confidence through memorizing and reciting poems. It can also provide an easy way for children to remember a lesson or value.
Claim: It can also provide an easy way for children to remember a lesson or value.
Answer: {"label": "no", "decontext": "Poetry can provide an easy way for children to remember a lesson or value."}

Example 2:
Context: Yes, ancient societies had concepts of rights. The concept of rights first appeared in the theory of natural law which existed in the state of nature. In this state, people enjoyed certain rights sanctioned by natural law.
Claim: In this state, people enjoyed certain rights sanctioned by natural law.
Answer: {"label": "no", "decontext": "In the state of nature, people enjoyed certain rights sanctioned by natural law"}

Example 3:
Context: The ancient Greeks had some concept of human rights, although there is no single word in classical Greek that captures the sense of "rights" as it is used in modern political thought. However, Greek customs and institutions provided protection to private property unique in the ancient world, instilling a strong sense of equality. The idea of human rights spread quickly from Babylon to Greece and eventually Rome, where the concept of "natural law" arose.
Claim: The idea of human rights spread quickly from Babylon to Greece and eventually Rome, where the concept of "natural law" arose.
Answer: {"label": "yes", "decontext": "NA"}

Your Turn:
Context: [CONTEXT]
Claim: [CLAIM]
Answer:

Table 24: Decontextualization prompt for GPT-4.

We are creating a news article (one paragraph) in the style of The New York Times. We will give you a claim that must be covered when writing your article. All information in the claim must be supported by weaving together various pieces of evidence within the text. That is, the claim should not be directly supported by using one sentence from the article. The generated article should be around 140 words.

Claim: [CLAIM]
Article:

Table 25: Supporting document generation prompt for the simplified data generation method C2D-SIMP.

You are presented with a claim and an article that fully support the claim. You task is to minimally modify the article with the following requirements:

1. The modified article no longer fully supports the claim. Some (but not all) statements in the claim should be supported by the modified article.
2. The edited article looks close to the original claim.
3. The edited claim article should have the similar length with the original article.

The followings are the type of revisions you can use to revise the article:
- Entity revision: An entity (like a person, place, organization, etc.) from a claim is being edited or not mentioned in the revised article.
- Number revision: A number from a claim is being edited or not mentioned in the revised article.
- Attribute revision: A syntax unit (either a word, phrase or clause) that modifies a noun is being edited or not mentioned in the revised article.
- Predicate revision: A main content verb or content like adverbs that closely relate to the verb is being edited or not mentioned in the revised article.

Claim: [CLAIM]
Article: [ARTICLE]

Please respond in a JSON format: {"revision_type": ..., "revised_article": ...}.

Table 26: Nonsupporting document generation prompt for the simplified data generation method C2D-SIMP.

Document:
[DOCUMENT]

Consistent Summary:
[CONSISTENT_SUMMARY]

Given the document and consistent summary above, generate 10 slightly modified versions of the summary such that the modifications introduce a factual inconsistency. For example, you can modify a number, date, or entity, and negate or modify a statement. Here are some rules to follow:
- Each modification should change at most 3-4 words from the original summary, and keep the rest the same.
- Each modification should change a different part of the original summary.
- Your modifications should be challenging to detect: modify minimally while still introducing a factual inconsistency.
- The factual inconsistency you introduce should be subtle. For example, if you replace an entity, make sure you replace it with another entity from the document.
- Each modification should start with "[FIRST_THREE_WORDS] [...]", and end with "[LAST_THREE_WORDS]"

Please respond in a JSON format with the following structure:
{"inconsistent_summaries": ["First inconsistent summary", "Second inconsistent summary", ...]}

Table 27: Nonsupporting document generation prompt for the simplified data generation method D2C-SIMP. The prompt is adapted from SummEdit (Laban et al., 2023).

Determine whether each of the provided claims are consistent with the corresponding document. Consistency in this context implies that all information presented in a claim is substantiated by the document. If not, it should be considered inconsistent.
Document: [DOCUMENT]
Claims: [CLAIM]

Claims are displayed with sentence indices. Please evaluate each claim's consistency with the document by responding with either "yes" or "no" in the JSON format: {"[1]": ..., "[2]": ..., ...}.
Answer:

Table 28: Prompt for predicting the factuality labels of all claims in a response for a provided document. This is used mainly for text summarization where multiple summary sentences share the same document.