ECAI 2025
I. Lynce et al. (Eds.)

doi:10.3233/FAIA251296

SARSum: A Relevance and Comprehensiveness-Aware Abstractive Summarization Dataset for Suspicious Activity Reports

Jean V. Alves[a,*], Javier Liébana[a], Hugo Ferreira[a] and Pedro Bizarro[a]

[a]Feedzai

Abstract. Existing benchmarks that evaluate the ability of Large Language Models (LLMs) to summarize rely primarily on measuring a summary's lexical similarity to a reference or on assessing whether its claims are factually consistent with the source document. These approaches fail to account for a summary's comprehensiveness — the extent to which it captures important information, and relevance — the extent to which unessential elements are omitted. To bolster comprehensiveness and relevance evaluation in high-stakes domains, we propose SARSum, a dataset tailored to evaluate the summarization of notes taken by anti-money laundering (AML) analysts during the process of preparing a Suspicious Activity Report (SAR), a document filed by financial institutions to alert law enforcement about suspicious transactions or activities, where omission of key details can be extremely costly. To the best of our knowledge, SARSum is the first comprehensiveness and relevance-aware summarization dataset: each of the 2,000 sets of notes is accompanied by the key facts that must be retained in an ideal summary, along with 30 different summaries spanning six levels of information selection quality, created by either omitting key facts or introducing irrelevant information. These resources allow practitioners to evaluate not only a summary's relevance and comprehensiveness, but also the ability of automatic metrics to assess them. These instances are generated using a variety of LLMs to rephrase templates approved by an AML expert, and we empirically verify that the resulting instances are highly abstractive and varied. While SARSum addresses a specific domain, the novel inclusion of key facts and a reference set with known levels of quality represents a crucial step with potential for broader application across high-stakes scenarios. These elements enable the use of techniques such as natural language inference and question-generation/question-answering to evaluate relevance and comprehensiveness.

1 Introduction

Automatic text summarization has become a vital tool for managing the overwhelming volumes of textual information in many domains. To generate an effective summary, a system must extract the essential content from the source document, omitting less important details and minimizing redundancy. Initial automatic summarization systems were extractive, producing summaries by selecting and joining excerpts taken verbatim from the source, resulting in disjointed sentences and poor readability. In contrast, abstractive summarization systems digest, paraphrase and condense information to create more human-like summaries [33, 18]. Abstractive text summarization evolved drastically with transformer based approaches such as BART [21], which perform highly in benchmarks [10, 40], but require extensive training datasets to adapt to new settings [12]. In contrast, LLMs have shown remarkable results in zero and few-shot tasks across a wide range of domains [7, 1, 34, 8], allowing for different summarization styles with simple prompt changes [12]. In spite of their ability to generate highly fluent text, LLMs are known to hallucinate [16, 17], and may fail in selecting the subset of important information from the source. Robust evaluation of LLM generated summaries is thus necessary for widespread trust and adoption of these models, particularly in high stakes scenarios such as medicine, finance or law. Thorough evaluation of automatically generated summaries must place significance on both *factual consistency* (i.e., the requirement that all information be inferable from the source document), and the quality of *information selection*, which we decompose into two criteria: *relevance* (i.e., the requirement that no unessential information is included in the summary), and *comprehensiveness* (i.e., the requirement that no key points/information nuggets in the source are omitted). These measures can be quantified as

$$\begin{aligned}\text{factual\_consistency} &= \#\text{consistent\_facts}_{\text{summary}}/\#\text{facts}_{\text{summary}},\\ \text{relevance} &= \#\text{key\_facts}_{\text{summary}}/\#\text{facts}_{\text{summary}},\\ \text{comprehensiveness} &= \#\text{key\_facts}_{\text{summary}}/\#\text{key\_facts}_{\text{source}}.\end{aligned}$$

In spite of the importance of these three measures, the most popular summarization datasets – CNN/DM [14] and XSum [26] – provide practitioners only with news article and reference summary pairs. While these allow users to measure how similar candidate summaries are to the provided ones, it is impossible to derive a rigorous measure of the reference's factual consistency, comprehensiveness, or relevance. Other datasets, such as FRANK [27] and QAGS [38], allow for factual consistency evaluation, as each sentence in the summaries is annotated as factually consistent or inconsistent with the source document. However, these two datasets place no importance on either comprehensiveness or relevance, as there is no annotation for which pieces of information from the source document must be retained in the summary. An additional issue stems from the fact that most of the commonly used benchmarks in recent work focus on news article summarization, neglecting to consider high-stakes scenarios such as medicine, finance, or law, where the omission of important information can have catastrophic consequences. These settings often involve processing documents with unique structures and technical

[*] Corresponding Author. Email: jean.alves@feedzai.com.

Please contact the corresponding author for any appendices or supplementary material mentioned in the paper.

jargon, thus necessitating domain-specific benchmarks.

To tackle these issues and bolster thorough evaluation of both comprehensiveness and relevance in automatic summarization, we propose SARSum: a dataset pertaining to the summarization of anti-money laundering (AML) notes, a high-stakes scenario in which errors can lead to incorrect reporting of financial crimes. Real AML analysts write and summarize these notes in the process of creating a Suspicious Activity Report (SAR) [3], which is then sent to regulatory or law enforcement agencies, leading to legal action. In this dataset, the source document is a set of chronologically ordered, synthetically generated notes regarding the activities of a business or individual under suspicion of money-laundering. These notes are highly structured, containing a short description of a subject's activity, a deliberation on whether it constitutes suspicious behavior, and a more thorough detailing of the observed events. An automated summarization system should produce a cohesive paragraph relaying the key suspicious activities of the investigation's subject.

Our dataset places special focus on the evaluation of comprehensiveness and relevance: for each source document, SARSum contains **(a)** a set of key facts, sentences which convey a critical piece of information, allowing practitioners to quantify both relevance and comprehensiveness (see Table 1); **(b)** a total of 30, factually correct, distinct summaries spanning 6 distinct levels of information selection quality, containing, from highest to lowest: **(1)** five perfect summaries, containing only the key facts; **(2)** five summaries containing all key facts with an additional piece of irrelevant information taken from the source document; **(3)** five summaries which omit the same key fact; **(4)** five summaries derived from (3) with an additional nonessential excerpt; **(5)** five summaries which omit two key facts, including the one missing in (3); and, finally **(6)** five summaries derived from (5), containing additional inconsequential information. This ranking assumes that the inclusion of a non-crucial piece of information is a less costly error than the omission of an important fact. The set of key facts, paired with the high number of summaries with known levels of quality, allows for the thorough evaluation of metrics which quantify both relevance and comprehensiveness [23, 11]. For additional utility, the key named entities (i.e., Companies, Dates, etc.) included in each component of the dataset are enclosed within curly braces (e.g., On {May 12, 2022} [...]), allowing for the evaluation of metrics involving named entity recognition [38, 9].

We evaluate SARSum by comparing it to the two most commonly used summarization datasets, XSum and CNN/DM, placing significant focus on evaluating the abstractiveness of the summaries within each dataset, based on measures used in previous work [13, 9], as well as our own. Due to the highly specific domain of SARSum, we also perform an analysis of the vocabulary distribution and the degree of compression – the ratio between source document and summary length – and how these compare to the news summarization domain.

In summary, our contributions are:

- SARSum: a dataset geared towards the evaluation of relevance and comprehensiveness in automatic abstractive summarization, containing not only 5 reference summaries and the key facts contained within them, but also 25 summaries with known levels of sub-par information selection, see Section 3.
- An analysis of the abstractiveness, vocabulary and compression of SARSum summaries when compared to the two most commonly used datasets: XSum and CNN/DM, see Section 4.
- Extensive discussion on the unique characteristics of SARSum (key facts, annotated named entities and multiple reference summaries) and their usefulness in abstractive summarization evaluation, see Section 2 and 4.

2 Related Work

Over the course of this Section, we provide an overview of the current state of research in automatic summary evaluation. We first cover publicly available summarization datasets, concluding that, to the best of our knowledge, no currently available resources serve as a sufficient benchmark for relevance and/or comprehensiveness evaluation. We also cover several automatic summarization metrics and how SARSum is useful in assessing their quality.

2.1 Summarization Datasets

Most work on benchmarking abstractive summarization is based around two highly popular summarization datasets: CNN/DM [14] and XSum [26], both pertaining to news article summarization. CNN/DM consists of news articles and associated human-created bullet-point summaries. XSum contains single-sentence summaries written by journalists as introductions to the news articles they precede. Given their widespread use to evaluate abstractive methods, it is crucial that the references be truly abstractive. Durmus et al. [9] evaluate the abstractiveness of both CNN/DM and XSum, concluding that XSum is much more abstractive; for example, over 10% of summary sentences in CNN/DM can be formed by deleting words in a source sentence, while none in XSum can. As a result, benchmarks such as Realsumm [5] and SummEval [10]—which provide human judgments for systems trained on CNN/DM—are expected to contain mostly extractive summaries, as even abstractive methods trained on CNN/DM produce highly extractive outputs [9]. XSum also has a critical flaw: 73% of these references, scraped from online metadata, contain information not found in the article itself [25]; other scraped datasets, such as Newsroom [13], may share this issue.

FRANK [27] and QAGS [38] both provide human annotations for the factuality of each sentence within summaries extracted from CN-N/DM and XSum. However, they do not provide any ground truth for what information must be retained in the summary, meaning that a summary containing a single fact from the source (relevance aside) will still be labeled perfectly factually consistent (see Table 1).

While the key issue of automatic summarization is the identification of the information that should be included in the summarized text, to the best of our knowledge no prior abstractive summarization dataset provides annotations identifying which facts present in the source must appear in the summary. Furthermore, all of the aforementioned datasets involve summarizing news articles, making SARSum a valuable contribution, not only in being a rigorous relevance- and comprehensiveness-aware abstractive summarization dataset, but also in diverging from the news summarization task and enabling evaluation in a different, high-stakes domain.

2.2 Automatic Summary Evaluation

While human evaluation is used in several works to rank the quality of different summarization systems [10, 25, 26], the process of collecting human evaluations of summaries is expensive, leading many researchers to resort to automatic summary evaluation metrics. In this section we describe several of these metrics, and how SARSum serves as an excellent benchmark for their efficacy, particularly when it comes to the evaluation of relevance and comprehensiveness.

2.2.1 Reference-Based Similarity Metrics

Reference-based metrics are the most common for both abstractive and extractive automatic summary evaluation. These involve quanti-

	XSum & CNN/DM	FRANK & QAGS	SARSum
Factual Consistency	N/A	$\#\text{consistent\_facts}_{\text{summary}} / \#\text{facts}_{\text{summary}}$	N/A
Relevance	N/A	N/A	$\#\text{key\_facts}_{\text{summary}} / \#\text{facts}_{\text{summary}}$
comprehensiveness	N/A	N/A	$\#\text{key\_facts}_{\text{summary}} / \#\text{key\_facts}_{\text{source}}$

Table 1. Possible Factual Consistency, Relevance, and Comprehensiveness measures within each dataset.

fying similarity to a gold-standard summary. Several metrics measuring n-gram overlap between a generated text excerpt and one or more references have been proposed as a means to automatically evaluate natural language generation [22, 28, 36, 4]. Among these, the most common is ROUGE [22]. In 2021, two-thirds of papers on automatic summarization published in NAACL and ACL measured summarization performance only in terms of ROUGE [19]. Due to these metrics' sensitivity to changes in phrasing, practitioners are strongly encouraged to collect or synthetically generate multiple summary references, which may be prohibitively expensive. To ensure that these metrics can be thoroughly evaluated in our dataset, we provide users with a set of 30 summaries per instance, comprised of 6 subsets of 5 different paraphrasings of a summary with a given quality level, thus allowing practitioners to assess how these metrics correlate with our 6 levels of quality by using up to 5 references.

However, as these metrics measure only semantic similarity, they fail to provide a score for specific desiderata (e.g. factual consistency, relevance, comprehensiveness), leading to the development of automatic metrics that quantify different dimensions of summary quality.

2.2.2 Factual Consistency, Comprehensiveness, and Relevance Evaluation

Nugget-based Evaluation Introduced by Voorhees [37], the nugget evaluation methodology emphasized identifying "nuggets", or key facts, relevant to a good answer. However, these systems are designed primarily to locate relevant spans across large corpora, not to rephrase them into fluent summaries—a core requirement in abstractive summarization. Because rephrasing is not required, methods typically rely on n-gram based matching [29, 24], inheriting the limitations discussed in the previous section.

More recently, LLMs have been applied to nugget extraction and to evaluating the relevance of retrieved information [2, 30]. While related to relevance and comprehensiveness, these resources are not suitable for our use case, where the key challenge is to verify that very short, highly abstractive summaries include all important facts and omit irrelevant information. By contrast, nugget-based evaluation emphasizes nugget generation/extraction and the relevance of a document in the context of retrieval-augmented generation.

Natural Language Inference Several works frame the task of factual consistency evaluation as a Natural Language Inference (NLI) [6] problem. NLI involves classifying the relationship between two pieces of text, a premise and a hypothesis, into one of three categories: the hypothesis is supported by (entailed by) the premise, contradicts it, or is neutral with respect to it. NLI models have been adopted in factual consistency evaluation by checking for entailment between the source-document and the summary, or between pairs of sentences from each document, in datasets with factual consistency annotations such as FRANK and QAGS [25, 15, 20]. While the use of entailment has been extensively researched for factual consistency assessment, its applicability to relevance or comprehensiveness evaluation is rarely considered. SARSum allows for the application of NLI to the evaluation of summary comprehensiveness, by measuring the entailment between a summary and the key facts. If a key fact present in the source document is classified as not entailed by the summary, one can infer said fact is not present in it. A similar approach has been explored using LLMs by Liu et al. [23], who propose generating a list of key points from the source text, and then asking the LLM whether the summary contains said point. Conversely, relevance can be evaluated by decomposing the summary into sentences, verifying whether each sentence is entailed by a key fact. Our dataset serves as an important resource in the evaluation of these metrics, when applied to comprehensiveness and relevance, by providing a ground truth for the key facts. This allows for the validation of key fact extraction from the source document, and entailment assessment between the summary and ground truth key facts.

Question Generation and Question Answering Question Generation and Question Answering (QG-QA) approaches are based on the notion that a factual summary should provide the same answers as the source document to a set of relevant questions. Durmus et al. [9] and Wang et al. [38] propose two similar reference-free approaches. Named entities are extracted from a source document using a pretrained model and treated as answers to potential questions. The Question Generation (QG) model receives both the answer and the source and is trained to generate the corresponding question. At test time, questions are created conditioned on entities extracted by the same model, and a Question Answering (QA) model answers them either from the source or the summary. The similarity of the answers is then used as a measure of factual consistency. The work of Scialom et al. [32] focuses on comprehensiveness by generating questions from the source and training a query weighter to determine which ones are "important" and should be answered by the summary.

Our dataset supports these methods in two ways. First, named entities in the source and summaries are enclosed in curly braces, allowing practitioners to evaluate extraction accuracy. Second, the availability of key facts enables generating questions with guaranteed relevance by taking a known key fact as the answer. This approach is used by Liu et al. [23], who use key facts extracted by an LLM to generate questions. SARSum allows practitioners to check the relevance of generated questions, evaluate QA model quality on highly relevant ones, and measure comprehensiveness by verifying whether all key-fact questions can be answered from the summary.

The 30 summaries with 6 known quality levels also make it possible to test whether automated metrics correctly rank summaries based on relevance and comprehensiveness, while assigning similar scores to paraphrases of the same quality. Since most evaluations of these metrics were done on news datasets, and relative rankings vary significantly across domains [20, 15, 39, 11], this highlights the importance of SARSum as a novel resource for evaluating not only abstractive summarization methods but also the automated metrics proposed to analyze their quality.

3 Dataset Generation

In an AML investigation, experts are tasked with analyzing the activities of a business or individual, in order to uncover patterns of behavior that point to potential financial misconduct. During the course of an investigation, AML analysts often compile a large set of notes regarding specific activities carried out by the subject, ranging from routine restocking purchases to attempts to move funds while circumventing regulations. The summarization of these notes into a cohesive narrative is a crucial step in developing a Suspicious Activity Report, a document which is then filed to financial crime authorities. In our dataset, each instance contains a set of notes to be summarized, the set of key facts contained within said notes, and 30 summaries with varying levels of quality, as detailed in Section 1. Each of these components is generated based on randomly selecting one out of 8 templates, which were all deemed realistic by a real AML analyst. For each component, we now describe the generation process in extensive detail. The code used to generate our dataset, the dataset, and an Appendix, are provided [35].

3.1 Source Document

The source document is a set of notes pertaining to the behavior of an individual or business. These are short paragraphs focused on a specific business activity, and an AML analyst's judgment on whether it constitutes suspicious behavior. For each note, the template contains two values: the note structure – a list of tuples containing a segment of text and the probability that said excerpt will be included in the note – and the probability that the note itself will be included in the final set of notes. An example of the template for a note is given in Figure 1. All notes must start with a "Pattern Identified:" and "Decision:" markers, where the AML analyst succinctly describes the observed behavior and whether or not it is suspicious.

Following the sampling of the note's segments, the values for each named entity placeholder are sampled from a synthetic data pool and introduced within the text, while keeping them enclosed in curly braces. This bracket annotation of the named entities is essential to guarantee that these values are not altered in the following steps. After replacing the placeholders with the values, the next step is the rephrasing of the note by using an LLM. First, a rephrasing prompt is created by sampling from various possible text segments which define the desired tone and rules which the LLM must obey (i.e. "you may not change the values enclosed within '{' and '}' characters" or "you must keep the "Pattern Identified:" and "Decision:" tokens within your response."). This text is then rephrased using the selected LLM, and this process may be repeated on the output up to 2 more times. The variable rephrasing prompt and consecutive rephrasing steps ensure that there will be significant lexical diversity within the generated samples, which we analyze in Section 4. For additional variability on the consecutive rephrasing steps, the prompt also includes, with a given probability, the entire conversation history, thus ensuring that it strays farther from the phrasing in the original template. Finally, the use of four different LLMs to generate samples also contributes to further diversity in the phrasing of these notes. In our work, we used Claude 3 Haiku, Gemini 1.5 and 2.0 Flash, and GPT-4o-mini. At each rephrasing step, we verify that the LLM's output note contains the exact same placeholders. If the model fails to obey this restriction, the generation is retried up to 3 times and is aborted if the final attempt is not successful. A more thorough description of the rephrasing prompt sampling process, as well as a few example prompts, are available in the Appendix. Figure 2 shows an example of a note generated from the template shown in Figure 1.

"Pattern Identified: Payment for consulting services.
Decision: Not suspicious.
{Jerry Edwards} paid ${442000} for consulting services related to their profession following a wire transfer on {June 20, 2021}"

Figure 2. Example note, generated by rephrasing the template in Figure 1

.

3.2 Key Facts

Each of the eight templates also contain the key facts related to the notes described previously. Key facts represent information that must be contained in a complete summary. These facts are represented by a single sentence, following the same structure as the templates for the notes themselves. For example, a fact pertaining to the information present in the note shown in Figure 1 is "{{{INDIVIDUAL}_0}} payed ${{{AMOUNT}_0}} for consulting services.". Each template contains around 10 key facts. In order to provide each fact in a variety of different phrasings, we conduct the same rephrasing step mentioned in the previous subsection. Each instance contains up to five different phrasings of each key fact, as some key facts are too succinct to rephrase in 5 distinct manners. This allows for testing of entailment/QG-QA methods with robustness to paraphrasing.

3.3 Summaries

The summaries are constructed in the same manner as the aforementioned notes, but contain only a cohesive paragraph summarizing the observed suspicious activities, accompanied by a final recommendation on the necessity of further investigation. Within each template, there are 6 different summaries, corresponding to each level of quality mentioned in Section 1. The templates corresponding to the summaries with a perfect score were also validated by an AML analyst, who deemed these to be high quality summaries.

4 Data Analysis

In this Section, we analyze several key characteristics of our dataset, first focusing on the abstractiveness of SARSum's summaries. We then assess the diversity across the generated instances. Finally, we

```
Note(note_structure=[
("Pattern Identified: Payment for consulting services.\nDecision: Not suspicious.\nFollowing the initial wire transfer,
     on {{{DATE}_3}}, {{{INDIVIDUAL}_0}} made a payment of ${{{AMOUNT}_7}} for consulting services related to their
     profession. ", 1),
("The services were provided by a reputable firm within {{{COUNTRY}_0_{CITY}_0}}, {{{COUNTRY}_0}}, and the transaction
     was in line with the market cost for such services.", p_optional_segments),
], probability=p_optional_notes)
```

Figure 1. Example Note Template. Values within curly braces are placeholders which are then replaced by random values. The second segment is optional - it can be removed without omitting essential information. The entire note is added to the set of notes in an instance with probability "p_optional_notes"

	n-gram measures (%) (↑)			Extraction measures (%) (↓)		
Dataset	**1-grams**	**2-grams**	**3-grams**	**word**	**span**	**sentence**
SARSum	37.9 ± 9.4	76.7 ± 9.3	88.1 ± 6.4	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
XSum	38.7 ± 15.8	84.8 ± 12.0	96.1 ± 6.7	0.03 ± 2.0	0.02 ± 1.466	0.01 ± 0.704
CNN/DM	14.7 ± 9.2	52.1 ± 18.5	70.8 ± 20.4	11.4 ± 29.2	2.7 ± 10.6	1.6 ± 8.8

Table 2. Abstractiveness scores for SARSum, XSum and CNN/DM. "n-gram measures" indicate the percentage of novel n-grams. "Extraction measures" indicate the percentage of summary sentences generated with each extraction strategy. Intervals represent standard deviation.

verify that each of the 5 reference summaries within an instance's set are sufficiently different from one another, conducting the same analysis for the set of Key Facts included in each instance. Additionally, we exemplify how SARSum allows for more in depth evaluation of factuality metrics, due to its annotated named entities. All the code used in this Section is included in the supplementary material.

4.1 Abstractiveness

As previously stated, current work places a significant focus on abstractive summarization methods. Abstractiveness is a desirable trait, as human created summaries are also abstractive in nature [33] – humans often paraphrase and restructure ideas, condensing one or more sentences into more concise text, while keeping its fluency. Extractive methods, on the other hand, involve directly extracting relevant spans of text, concatenating them into a final summary, resulting in a fragmented, less readable summary [33].

As our summaries are obtained via paraphrasing a pre-determined template with one of four LLMs, we expect them to be highly abstractive. We first quantify abstractiveness by defining a series of metrics which capture lexical variability and sentence-to-sentence extraction, and then compare our dataset's characteristics to the two most popular summarization datasets, both pertaining to news article summarization: XSum [26] and CNN/DM [14].

4.1.1 Quantifying Abstractiveness

Abstractive summarization includes paraphrasing, restructuring of ideas, and condensation of larger spans of text into novel sentences. Consequently, abstractiveness measures often focus on the degree of lexical overlap between the source text and the generated summary, classifying a summary as "more abstractive" the less overlap exists between the source text D and the generated summary S. The work of Grusky et al. [13] proposes two different measures of textual overlap between the summary and the source document. These measures are based on *extractive fragments*, $\mathcal{F}(D, S)$, spans of text which are present in both the source and the summary.

Extractive Fragment Coverage The coverage measures the ratio of words in the summary belonging to an extractive fragment.

$$\text{Coverage}(D,S) = \frac{1}{|S|} \sum_{f \in \mathcal{F}(D,S)} |f|. \tag{1}$$

This measure, however, can be drastically inflated by extractive fragments of a single word. The authors concede that a summary using similar wording to the source document, but drastically condensing information, would still score highly in terms of coverage. In our use-case, this measure may also be inflated by the presence of extremely common domain-specific terms such as "money laundering", "business activities", etc. As such we propose redefining this metric as

$$\text{Coverage}_k(D,S) = \frac{1}{|S|} \sum_{f \in \mathcal{F}(D,S)} |f| \, \mathbb{I}\big(|f| \geq k\big). \tag{2}$$

In this manner, we can calculate the coverage within a given summary while taking into account only spans of text with length equal to, or larger than k, as represented by the indicator function $\mathbb{I}$. Note that since each instance within the SARSum dataset contains 5 perfect summaries, we will calculate the Coverage score associated with each instance as the mean of the individual Coverage scores of each of the summaries. This procedure will be followed for all metrics mentioned henceforth.

Extractive Fragment Density To tackle the issues with the Coverage metric, the authors propose a Density metric, which is similar to the coverage metric, using the square of the fragment length. We again expand this definition to obtain

$$\text{Density}_k(D,S) = \frac{1}{|S|} \sum_{f \in \mathcal{F}(D,S)} |f|^2 \, \mathbb{I}\big(|f| \geq k\big). \tag{3}$$

Note that this metric assigns exponentially higher weight to lengthier extractive fragments, thus curbing the problems with the Coverage metric. However, in a highly abstractive summary, we still expect most of the extractive segments to be comprised of single words which are present in both D and S. Consequently, we will also calculate the density considering only extractive fragments of length $\geq k$.

Sentence-based Extraction Measures Following the work of Durmus et al. [9], we also evaluate the level of extractiveness of a given summary on a sentence-by-sentence basis. These metrics involve assessing if all of the tokens in a given summary sentence can be traced to a single sentence within the source document. The authors consider three possible *extraction methods*: **Sentence Extraction** – the summary sentence is identical to a source sentence; **Span Extraction** – the summary sentence is a sub-string of a source sentence; and **Word Extraction** – the summary sentence is entirely comprised of words present in a source sentence. We define the score associated with each type of sentence-to-sentence extraction by calculating the percentage of summary sentences formed by said extractive method.

n-gram extractiveness evaluation The work of Durmus et al. [9] also calculates the percentage of n-grams present in the summary that are not present within the source text, for $n \in \{1, 2, 3\}$. While this measure is somewhat similar to the extractive coverage metric, we also calculate these values for comprehensiveness.

Compression To measure the amount of information compression from the source document to the summary, we calculate the ratio between the lengths of the source document and the reference summaries, such that $\text{Compression}(D, S) = |D|/|S|$. The compression of the reference summaries is given by 2.4 ± 0.5 in SARSum, 18.7 ± 22.5 in XSum, and 15.6 ± 9.6 in CNN/DM.

Vocabulary Distribution Due to the fact that our summarization task is specifically designed for the development of Suspicious Activity Reports, the summarization style is expected to be very different from that of news articles. One aspect where this disparity is evident is the vocabulary used within these domains. In Figure 3, we display

	Coverage (↓)			Density (↓)		
Dataset	**Coverage$_1$**	**Coverage$_2$**	**Coverage$_3$**	**Density$_1$**	**Density$_2$**	**Density$_3$**
SARSum	0.63 ± 0.09	0.26 ± 0.10	0.14 ± 0.07	1.26 ± 0.42	0.89 ± 0.43	0.65 ± 0.39
XSum	0.61 ± 0.16	0.14 ± 0.13	0.04 ± 0.08	0.82 ± 0.43	0.35 ± 0.44	0.15 ± 0.41
CNN/DM	0.85 ± 0.09	0.47 ± 0.19	0.32 ± 0.22	3.01 ± 3.60	2.63 ± 3.6	2.33 ± 3.74

Table 3. Extractive fragment Coverage and Density values for SARSum, XSum and CNN/DM.

the top 10 most frequent words and their relative frequency within the SARSum, XSum and CNN/DM datasets. For a measure of the distribution's peakedness, we use Shannon's Entropy.

Named Entities Note that due to the nature of these notes, a significant part of the text is comprised of named entities, such as addresses, names of individuals and companies, monetary amounts, among other key fields that may not be altered when rephrasing the text. Consequently, if these entities cover a large amount of the source document's vocabulary, we expect the same to be true for the summary, thus imposing a restriction on the amount of abstractiveness. To measure the extent to which the source document and the summary are composed of these inalterable entities, we use the Coverage$_1$ and Density$_1$ metrics. For the source document D, we obtain 0.18 ± 0.06 and 0.86 ± 0.87, respectively, while for the reference summaries S, we obtain 0.17 ± 0.07 and 0.96 ± 1.16, respectively.

4.1.2 Analysis of Abstractiveness

Observing the results shown in Table 2 and Table 3, we can observe that, in line with previous work, the CNN/DM dataset is drastically less abstractive than XSum, with nearly 30% of 3-grams in the summary being present in the article. This is especially noticeable in the Extraction Measures, where SARSum scores zero and XSum scores almost zero in all extraction measures. While XSum scored zero on these metrics in the work of [9], we believe this disparity comes down to differences in how extraction is counted for. For example, we convert all words to lower case prior to comparison, which may lead to an increased number of matches. Nonetheless, this indicates that most summary sentences in SARSum and XSum are formed by combining information from multiple source sentences or by paraphrasing them. In spite of the slight advantage on extraction-based measures, SARSum lags behind XSum on other metrics. A contributing factor is the highly structured text format, with reference summaries mostly following a chronological order, and always ending with a recommendation for further investigation into the activities of the company under observation, which can increase the chance of similar text spans between the source and references.

Focusing on SARSum and XSum, we also observe that while the value of Coverage$_1$ is very similar for both, the values for Coverage$_2$ and Coverage$_3$, as well as all the Density metrics are significantly higher for the SARSum dataset, with the disparity between the two datasets rising as k increases. This indicates that SARSum summaries have a much higher tendency to retain spans of text comprised of 2 or more words from the source document when compared to XSum. A contributing factor to this phenomenon is the nature of the vocabulary used in SARSum. In Figure 3 we demonstrate that the relative frequency of the most common words within our reference summaries is significantly higher than that of the news article summarization datasets. These not only contain words that are commonly used by themselves, such as "transactions" and "company", but also in phrases containing more than a single word, such as "money laundering" and "business activities". The other factor contributing to this pattern is the high frequency of named entities within the text. While some of these, like city or country names, are comprised of a single word, many are comprised of longer spans of text, such as company names (i.e. Chapman, Barnes and Vega) or addresses (i.e. XX Chillingham Road, Heaton, Newcastle, XXX XXX). In the paragraph "Named Entities" in Section 4.1, we demonstrate that these named entities are a significant part of both the summaries and the source document. Measuring the coverage and density metrics without considering the named entities results in much lower values for all metrics (Coverage$_1$ $= 0.46 \pm 0.05$, Coverage$_2$ $= 0.11 \pm 0.05$, Coverage$_3$ $= 0.03 \pm 0.03$, Density$_1$ $= 0.63 \pm 0.15$, Density$_2$ $=$ 0.28 ± 0.15, Density$_3$ $= 0.13 \pm 0.13$), with an especially notable decrease for the density at larger values of k. This indicates that these named entities inflate the value of the coverage and density statistics, without pointing to a more extractive writing style, as these spans of text, by their very nature, cannot be rephrased.

Finally, the compression of information in XSum and CNN/DM is 7.5 and 6 times higher, respectively, than that of SARSum. This shows that the source document is already highly informative text, being much more dense in content that must be retained in the final summary. This highly affects the abstractiveness of the reference summaries, as less of the information present within the source document can be discarded, resulting in a higher overlap between the tokens in the source document and the summary.

This analysis demonstrates that SARSum contains highly abstractive summaries, on par with the most commonly used abstractive

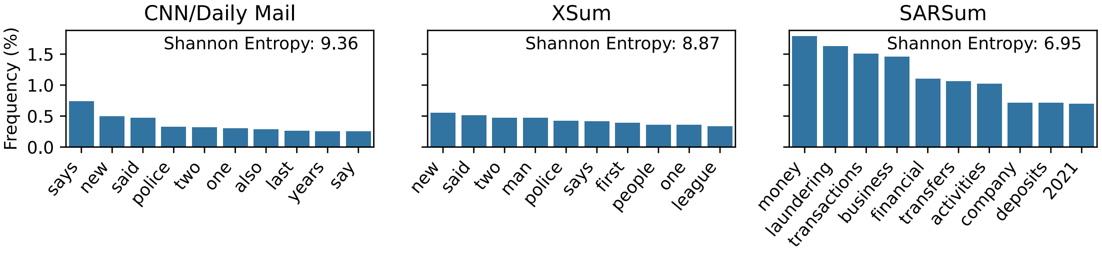

Figure 3. Relative frequency of 10 most common words within each dataset.

summarization dataset, XSum, while displaying several characteristics innate to the financial fraud detection domain, for which no current dataset exists, to the best of our knowledge.

4.2 Diversity of Summary References

For each instance, our dataset includes 5 different high quality reference summaries, allowing for testing metrics on their robustness to different phrasings. However, multiple reference summaries are only useful if these capture sufficiently different ways of phrasing the important information. Consequently, we evaluate the similarity between pairs of reference summaries for the same instance using the ROUGE-1 F1 score, which is given by

$$\text{Rouge1}_{\text{F1}}(S_1, S_2) = \frac{2\sum_w \min\{c_{S_1}(w),\, c_{S_2}(w)\}}{\sum_w c_{S_1}(w) + \sum_w c_{S_2}(w)}. \tag{4}$$

where w represents a unigram token, and $c_{S_1}(w)$ and $c_{S_2}(w)$ are its counts in each summary. We observe that, on average, reference summaries for the same instance have a Rouge1 F1-score of 0.72 ± 0.09. Knowing that these summaries must share named entities, we calculate the Rouge1 F1-score removing these, obtaining 0.66 ± 0.10. Considering the nature of the vocabulary in this high expertise domain, we believe that this constitutes significant lexical variability.

4.3 Diversity of Key Facts

Similarly to the summaries, each key fact associated with the instance is paraphrased in 5 different manners, in order to allow for the evaluation of entailment and QG-QA fact checking methods with robustness to different phrasings of the facts themselves. Again, it is desirable for these differently paraphrased texts to exhibit a high lexical diversity. Following a procedure similar to that of the evaluation of reference summary diversity, we find that phrasings for the same key fact exhibit a Rouge F1-score of 0.67 ± 0.06, constituting a significant amount of diversity in phrasing, especially when considering that key facts are sentences with around 10 words.

4.4 Inter-instance Diversity

Our dataset is generated based on 8 different templates. It is expected that, given sufficient sampling of the same template, very similar instances can be created. To measure the maximum similarity between two instances, we first calculate the similarity between the source documents, given by

$$Sim_D(i,j) = \text{Rouge1}_{\text{F1}}(D_i, D_j).$$

We then define the similarity between sets of summaries as the average similarity between the 25 possible pairs formed by sampling one of the summaries of i and one of the summaries of j, given by

$$Sim_S(i,j) = \frac{1}{25}\sum_{k=1}^{5}\sum_{l=i}^{5} \text{Rouge1}_{\text{F1}}(S_{i,k}, S_{j,l}).$$

The maximum similarity between i and j is then defined as

$$MS = \max_{i,j} \frac{1}{2}\big(Sim_D(i,j) + Sim_S(i,j)\big).$$

The highest MS between two instances is 0.72, indicating substantial difference between even the most similar of instances.

4.5 Named Entity Recognition

One key aspect of our dataset is the annotation of the relevant named entities within both the source document and the summary, being that these are enclosed within curly braces. The annotation of named entities is particularly useful for the development and analysis of QG-QA summary evaluation metrics, as discussed in Section 2. These metrics involve extracting named entities from either the source document or the summary, using them as the answers for a QG model. Our dataset allows practitioners to evaluate this key step in QG-QA evaluation by comparing the extracted entities with the annotated ones.

To conduct named entity recognition extraction, we follow a procedure similar to that of Wang et al. [38], using the pre-trained spaCy model *en_core_web_sm* (see https://spacy.io/models/en). For each set of reference summaries S, we will then have the set of ground truth named entities N, as well as the set of distinct named entities extracted by the spacy model E. We define a recall measure as $\text{Recall} = |E \cap N|/|N|$, and a precision measure as $\text{Precision} = |E \cap N|/|E|$. We can thus calculate an F1 measure by taking $\text{F1} = 2\cdot\text{precision}\cdot\text{recall}/(\text{precision}+\text{recall})$. We obtain $\text{F1} = 0.69 \pm 0.20$, where the interval represents a standard deviation. This value shows that this commonly used named entity extraction model is suboptimal at identifying the entities annotated within our dataset. Since the question generation step of many QG-QA approaches [9, 38, 31] hinges on adequate named entity extraction, this shows that our dataset poses a significant challenge for these metrics, and can serve to identify potential avenues for improvement.

5 Conclusion, Limitations and Future Work

In this work we propose SARSum, the first abstractive summarization dataset in the financial fraud and anti-money laundering domain, and, to the best of our knowledge, the first that accurately measures relevance and comprehensiveness by explicitly annotating key facts, and providing summaries with known relevance and comprehensiveness related errors. SARSum also allows for evaluation of named entity extraction, as all entities within each instance are enclosed in curly braces. We demonstrate that SARSum's summaries are highly abstractive and varied, making our dataset particularly relevant under the current LLM-focused paradigm of automatic summarization.

The main limitation of SARSum is its size, containing only 2000 instances. This is a low amount when compared with the most commonly used resources such as XSum and CNN/DM, which both boast hundreds of thousands of instances. The main reason for this is the cost associated with the generation of each instance, which involves tens of calls to an LLM's API, in order to paraphrase large amounts of text. However, the variability between instances in our dataset means that each instance is highly valuable. The value of each instance is further compounded by the existence of ground truth key facts, annotated named entities, as well as 30 summaries with known levels of quality per instance; offering new possibilities in the study of comprehensiveness and relevance.

In future work, we aim to create a highly granular benchmark of several LLM's performances within the SARSum dataset, evaluating not only the quality and reliability of LLM summarization in a high-stakes scenario, but also the performance of several automated metrics. This benchmark will include a thorough analysis of recent work in LLM-based evaluation, using SARSUM to assess relevance and comprehensiveness based metrics.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] M. Alaofi, N. Arabzadeh, C. L. Clarke, and M. Sanderson. Generative information retrieval evaluation. In *Information access in the era of generative ai*, pages 135–159. Springer, 2024.

[3] R. M. Axelrod. Criminality and suspicious activity reports. *Journal of Financial Crime*, 24(3):461–471, 2017.

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

[5] M. Bhandari, P. Gour, A. Ashfaq, P. Liu, and G. Neubig. Re-evaluating evaluation in text summarization. *arXiv preprint arXiv:2010.07100*, 2020.

[6] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[8] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[9] E. Durmus, H. He, and M. Diab. Feqa: A question answering evaluation framework for faithfulness assessment in abstractive summarization. *arXiv preprint arXiv:2005.03754*, 2020.

[10] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, and D. Radev. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 2021.

[11] M. Gao, J. Ruan, R. Sun, X. Yin, S. Yang, and X. Wan. Human-like summarization evaluation with chatgpt. *arXiv preprint arXiv:2304.02554*, 2023.

[12] T. Goyal, J. J. Li, and G. Durrett. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356*, 2022.

[13] M. Grusky, M. Naaman, and Y. Artzi. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. *arXiv preprint arXiv:1804.11283*, 2018.

[14] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

[15] O. Honovich, R. Aharoni, J. Herzig, H. Taitelbaum, D. Kukliansy, V. Cohen, T. Scialom, I. Szpektor, A. Hassidim, and Y. Matias. True: Re-evaluating factual consistency evaluation. *arXiv preprint arXiv:2204.04991*, 2022.

[16] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 2023.

[17] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

[18] H. Jing. Sentence reduction for automatic text summarization. In *Sixth applied natural language processing conference*, pages 310–315, 2000.

[19] J. Kasai, K. Sakaguchi, R. L. Bras, L. Dunagan, J. Morrison, A. R. Fabbri, Y. Choi, and N. A. Smith. Bidimensional leaderboards: Generate and evaluate language hand in hand. *arXiv preprint arXiv:2112.04139*, 2021.

[20] P. Laban, T. Schnabel, P. N. Bennett, and M. A. Hearst. Summac: Revisiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177, 2022.

[21] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[22] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[23] J. Liu, Z. Shi, and A. Lipani. Summequal: Summarization evaluation via question answering using large language models. In *Proceedings of the 2nd Workshop on Natural Language Reasoning and Structured Explanations (@ ACL 2024)*, pages 46–55, 2024.

[24] G. Marton. Nuggeteer: Automatic nugget-based evaluation using descriptions and judgements. *DSpace@MIT*, 2006.

[25] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*, 2020.

[26] S. Narayan, S. B. Cohen, and M. Lapata. Don‘t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

[27] A. Pagnoni, V. Balachandran, and Y. Tsvetkov. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.383. URL https://aclanthology.org/2021.naacl-main.383/.

[28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[29] V. Pavlu, S. Rajput, P. B. Golbus, and J. A. Aslam. Ir system evaluation using nugget-based test collections. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 393–402, 2012.

[30] R. Pradeep, N. Thakur, S. Upadhyay, D. Campos, N. Craswell, I. Soboroff, H. T. Dang, and J. Lin. The great nugget recall: Automating fact extraction and rag evaluation with large language models. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 180–190, 2025.

[31] T. Scialom, S. Lamprier, B. Piwowarski, and J. Staiano. Answers unite! unsupervised metrics for reinforced summarization models. *arXiv preprint arXiv:1909.01610*, 2019.

[32] T. Scialom, P.-A. Dray, P. Gallinari, S. Lamprier, B. Piwowarski, J. Staiano, and A. Wang. Questeval: Summarization asks for fact-based evaluation. *arXiv preprint arXiv:2103.12693*, 2021.

[33] H. Shakil, A. Farooq, and J. Kalita. Abstractive text summarization: State of the art, challenges, and improvements. *Neurocomputing*, page 128255, 2024.

[34] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[35] J. V. Alves, J. Liébana, H. Ferreira, and P. Bizarro. Sarsum: A relevance and comprehensiveness-aware abstractive summarization dataset for suspicious activity reports, Aug. 2025. URL https://doi.org/10.5281/zenodo.16887668.

[36] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[37] E. M. Voorhees. Overview ofthe trec 2003 question answering track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC2003)*, 2003.

[38] A. Wang, K. Cho, and M. Lewis. Asking and answering questions to evaluate the factual consistency of summaries. *arXiv preprint arXiv:2004.04228*, 2020.

[39] J. Wang, Y. Liang, F. Meng, Z. Sun, H. Shi, Z. Li, J. Xu, J. Qu, and J. Zhou. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*, 2023.

[40] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKeown, and T. B. Hashimoto. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57, 2024.