Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5B
D-70569 Stuttgart

Master thesis

Using Retrieved Augmented Generation for Question Answering with LLMs in the Cybersecurity Domain

Linnet Moxon

Studiengang: M.Sc. Computational Linguistics

Prüfer*innen: Dr. Agnieszka Faleńska
Prof. Dr. Sebastian Padó

Betreuer*innen: Dr. Agnieszka Faleńska
Johannes F. Loevenich, Thales Deutschland GmbH

Beginn der Arbeit: 01.11.2024
Ende der Arbeit: 15.06.2025

Erklärung (Statement of Authorship)

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und dabei keine andere als die angegebene Literatur verwendet habe. Alle Zitate und sinngemäßen Entlehnungen sind als solche unter genauer Angabe der Quelle gekennzeichnet. Die eingereichte Arbeit ist weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen. Sie ist weder vollständig noch in Teilen bereits veröffentlicht. Die beigefügte elektronische Version stimmt mit dem Druckexemplar überein.[1]

(Linnet Moxon)

[1]Non-binding translation for convenience: This thesis is the result of my own independent work, and any material from work of others which is used either verbatim or indirectly in the text is credited to the author including details about the exact source in the text. This work has not been part of any other previous examination, neither completely nor in parts. It has neither completely nor partially been published before. The submitted electronic version is identical to this print version.

Contents

1 Abstract

This work investigates the impact of Retrieved Augmented Generation on the performance of Large Language Models in a Question Answering scenario in the cybersecurity domain. 14 different context setups are built, based on the two data resources NIST and MITRE ATT&CK, and retrieved with tf-idf retrieval. The questions from Tihanyi et al. (2024) are prompted with and without context, and with and without answer options. The evaluation shows that the context only partially increases the performance of the three models meta.llama3-1-8b-instruct-v1:0, mistral.mixtral-8x7b-instruct-v0:1 and mistral.mistral-7b-instruct-v0:2. Further, there are differences observed in how well the models follow the prompt instructions, which clearly impacts the findings. In the future a more advanced retrieval method, different data resources for building the databases and a better alignment of the prompt length, could positively influence the results.

2 Introduction

Large Language Models (LLMs) become increasingly popular in the everyday private and working life. They are used for writing emails and scientific papers, improving the creativity of birthday cards, or collecting new recipes, they are also deployed in more critical areas, such as healthcare, jurisdiction and cybersecurity. LLMs are a type of language model that are trained on huge amounts of data (Yao et al., 2024), usually include a transformer architecture and billions of parameters (Minaee et al., 2024). They are a type of neural network that evolved out of statistical language models, early task-specific neural language models, and pre-trained task-agnostic language models (Minaee et al., 2024; Han et al., 2024). The transformer architecture allows for less training time due to its parallel architecture (Vaswani et al., 2017). Compared to earlier stages of statistical and neural language models LLMs have high natural language understanding skills, generation abilities (Minaee et al., 2024), and improved learning skills and can therefore be exploited for a wide range of tasks (Singh et al., 2024). They outperform other language models on few-shot

learning, meaning that they learn new tasks from only a few examples (Brown et al., 2020), and decompose complicated tasks into simpler components (Minaee et al., 2024). Abilities such as "reasoning, planning, decision-making, in-context-learning" (Naveed et al., 2023), data analysis, question answering, etc. are nothing that the models are trained on specifically, but what they inherit due to their large training data and the model architecture (Naveed et al., 2023; Han et al., 2024). Due to their advanced capabilities, LLMs are applied for versatile tasks. First, they are used for classic NLP tasks, such as classification tasks, for example sentiment classification, POS-Tagging (Part-of-Speech-Tagging), NER (Named Entity Recognition), and IR (Information Extraction) and text generation (Zhao et al., 2023). Further more, they can be used as embedding models (Lee et al., 2024), for IR (Zhao et al., 2023) and as multimodal models, where they are connected to modalities such as audio or images (Zhao et al., 2023).

Another common application scenario for LLMs is QA (Question Answering), where the users collect information with the help of the system by asking it questions. QA systems can be divided into closed domain and open domain QA systems (Allam and Haggag, 2012), and can be used for prompting questions with (Tihanyi et al., 2024) or without answer options (Lee et al., 2019). QA systems are not necessarily based on LLMs and have been used before the invention of LLMs. Nevertheless, LLMs inherit a wide range of knowledge and are therefore commonly used for QA (Hong et al., 2024).

Due to their knowledge and skills LLMs are employed in various domains, including sensitive areas such as healthcare, finances, jurisdiction and cybersecurity. In these domains their trustworthiness and reliability is of special importance, because human experts rely on information provided and decisions made by the LLMs. The current work takes a look at the deployment of LLMs as QA system in the cybersecurity domain. It investigates how well LLMs deal with answering questions from the cybersecurity domain and whether their performance in such a QA setup can be improved by augmenting them with additional domain specific knowledge, to provide human experts with reliable and truthful information.

The rest of the work is organized as follows: Section 3 gives background information

and discusses recent literature, leading to the definition of the Research Questions of the current work. Sections 4, 5, 6 and 7 examine the performance of LLMs on QA with and without context augmentation in the cybersecurity domain. Section 8 discusses the results, gives an outlook for future work and concludes the work.

3 Background and Related Work

This section provides background information and discusses recent literature regarding the topic of the current work. First limitations of LLMs will be discussed, followed by the detection of hallucinations in LLMs, LLMs in cybersecurity and RAG in QA. Afterwards, some retrieval and evaluation metrics will be described in more detail, leading to the specific Research Questions for the current work.

3.1 Limitations of LLMs

Despite their capabilities and application scenarios in various domains, LLMs also come with challenges and limitations. One is the outdated knowledge. LLMs can only use the knowledge coming from data, they were trained on, not real-time knowledge. The training data may be incorrect at a certain point (Han et al., 2024). Another problem with the training data is the huge size of it: the model might forget certain parts of the knowledge, especially when it is fine-tuned with more domain-specific data (Naveed et al., 2023). Another issue is privacy and security concerns. LLMs happen to leak personal information and are vulnerable to security attacks and misuse (Han et al., 2024; Naveed et al., 2023). Closely related to that problem is the problem of bias and fairness. Models learn patterns from their training data, which leads to ethical concerns, when the data contains harmful, discriminatory, and biased information (Han et al., 2024; Zhao et al., 2023). Another challenge, referred to as the black-box problem, is the lack of interpretability of LLMs (Naveed et al., 2023). Interpretability refers to the "extraction of relevant knowledge from a LLM about relationships either contained in the data or learned by the model", as defined by Singh et al. (2024), following Murdoch et al. (2019). The goal of

interpretability is to improve the LLM's performance by better understanding the underlying data and learning mechanisms, and to improve human confidence in the system (Wu et al., 2024). One related issue is the generation of hallucinations, where the model output contains incorrect, vague, ambiguous, or redundant information (Naveed et al., 2023; Han et al., 2024). Hallucinations refer to statements produced by LLMs, that seem reasonable and factual, but are ungrounded (Tonmoy et al., 2024), do not align with the input or world knowledge (Huang et al., 2023; Zhang et al., 2023b), contradict the context (Zhang et al., 2023b) or are cognitively and factually irrelevant (Ye et al., 2023; Xu et al., 2023). Detecting hallucinations and therefore making LLMs more interpretable is particularly important in cases where LLMs are used for decision making and/or in critical environments such as health care or cybersecurity. In these sensitive areas it is crucial to understand the models' outputs and therefore their decisions. The goal is to align the models with ethical standards and safety protocols (Wu et al., 2024). The detection and evaluation of hallucinations will be investigated in the next section.

3.2 Detection and evaluation of hallucinations in LLMs

Starting with the categorization of hallucinations in LLMs, Ye et al. (2023), Arteaga et al. (2024), and Huang et al. (2023) for example, distinguish hallucinations between factualness and faithfulness. Factualness refers to output or statements that contradict factual knowledge, whereas faithfulness refers to output or statements which do not align with the input. Zhang et al. (2023b) too, categorize hallucinations into input-conflicting hallucination, context-conflicting hallucination, and fact-conflicting hallucination. Hallucinations are especially hard to detect, because the model still produces fluent and coherent output, although it might contain wrong statements (Snyder et al., 2024).

For the detection, evaluation and and mitigation of hallucinations several methods are proposed in literature. In general, the methods can be divided into human evaluation, automatic evaluation with gold standard answers, and automatic evaluation without gold standard answers, where other LLMs or the target LLM's own confi-

dence are used to evaluate the output with regard to hallucinations. Model-based automatic evaluations and rule-based automatic evaluation methods are less costly than human-based evaluation metrics, but have their own shortcomings, for example LLMs being overly confident of their own answers (Zhang et al., 2023b; Huang et al., 2023). In the following paragraph a few exemplary automatic evaluation methods will be described. Arteaga et al. (2024) propose a hallucination detection method that outputs uncertainty estimates by using "the model's own confidence" (Arteaga et al., 2024) for rating the faithfulness and factualness of the output. Chen et al. (2024) build a metric based on EigenScore values for evaluating the self-consistency of the responses while maintaining the semantic information. A robust discriminator for detecting hallucinations is suggested by Chen et al. (2023), which is based on a bilingual question-answering dialogue dataset. Honovich et al. (2021) suggest an automatic evaluation metric for detecting factual inconsistency in question-answering dialogues. Instead of tokens they use span comparison and combine question generation and question answering for their method. Wei et al. (2024) tackle the challenge of detecting hallucinations without available gold standard answers, by using answers from another LLM as proxy for the gold standard answers. Their approach is to test the expertise of the of-the-shelf LLMs by measuring the untruthfulness, instead of the truthfulness.

QA is one of the NLP tasks where hallucinations play a crucial role. Different approaches are taken to detect hallucinations in the responses of models in QA settings. Farquhar et al. (2024) in their work focus on confabulations - a type of hallucination, where the produced statement is wrong and arbitrary and therefore caused by randomness. They propose semantic entropy for these claims that are hard to spot for a user, in order to ground the detection of this type of hallucination in meaning and less in specific terms (Farquhar et al., 2024). Snyder et al. (2024) detect artifacts in model responses that indicate that a statement contains hallucinations. Artifacts refer to certain pattern in the input, intermediate and output layer. These artifacts are then used to train another model to classify responses into hallucinations and non-hallucinations (Snyder et al., 2024).

3.3 LLMs in cybersecurity

Guaranteeing the trustworthiness of LLMs and detecting and mitigating hallucinations is especially important when deploying LLMs in critical or sensitive environments, such as healthcare, finances or cybersecurity. As other AI models, LLMs are used in different applications in the field of cybersecurity, which will be described in the following paragraph. According to Schatz et al. (2017) cybersecurity includes the "approaches and actions [...] followed by organizations and states to protect confidentiality, integrity and availability of data and assets used in cyberspace". Therefore "guidelines, policies and collections of safeguard, technologies, tools and training [are needed] to provide the best protection for the state of the cyberenvironment and its users" (Schatz et al., 2017).

Shafee et al. (2024) conduct a binary classification task and a NER task, using different LLMs to improve the extraction of cybersecurity information from text sources. The two tasks are crucial for filtering the relevant information and extracting lexical key items (Shafee et al., 2024). They investigate how well popular chatbots can be deployed for cybersecurity specific applications. Würsch et al. (2023) too investigate the performance of different LLMs on entity extraction. In their work, the model is used to extract knowledge from different text domains regarding cybersecurity to keep cybersecurity experts up to date regarding new attack and mitigation strategies. Gennari et al. (2024) mention that LLMs can be deployed for malicious activities, and understanding their abilities and functions is crucial to prevent such activities. The authors investigate whether LLMs can be trusted to "provide cybersecurity experts with reliable information" (Gennari et al., 2024). Therefore, opportunities and challenges of deploying LLMs for cybersecurity activities are examined, "how LLMs capture and apply cybersecurity knowledge in general" (Gennari et al., 2024), how specific tasks in the field of cybersecurity can be supported by LLMs and how cybersecurity experts can better evaluate LLMs' capabilities in the future.

Ji et al. (2022) develop a system where a LLM is supposed to respond to questions by extracting information from a cybersecurity knowledge base. Liu (2023) create a dataset for evaluating LLMs regarding their cybersecurity knowledge, as well as Tihanyi et al. (2024).

Since cyberattacks are getting more advanced through the exploitation of LLMs and other deep learning models, the defense of networks needs to be adjusted and improved accordingly. Loevenich et al. (2024) show how the defense of networks can be improved by combining different deep learning models, and make them work together as AI agents. In their work the LLM is used as an interface between the AI agents and human cybersecurity experts. The experts use the LLM to ask questions about the agent's work and their defense/attack techniques.

As already mentioned QA is a common LLM task, which can be used in a open-domain or closed-domain manner. When it is used in such critical areas as cybersecurity, the detection and mitigation of hallucinations is especially important to increase trustworthiness and interpretability. One method to decrease hallucinations in LLMs is Retrieved Augmented Generation (RAG). RAG in QA scenarios will be discussed in the next section.

3.4 RAG in QA

The objective of RAG is to provide a LLM with external additional (domain-specific) knowledge that is used as source for answering questions (Feldman et al., 2024; Jiang et al., 2023; Wu et al., 2023). This is needed because even large LLMs can not answer questions that exceed the scope of their training data and are not able to remember less prevalent knowledge in the training data (Es et al., 2024; Rau et al., 2024). For example, Loevenich et al. (2024) use RAG in their chatbot interface to enable human cybersecurity experts to ask questions regarding cybersecurity.
Belyi et al. (2024) suggest a method to detect hallucinations in RAG settings to "ensure reliability and accuracy of responses" (Belyi et al., 2024) of the LLM, based on BERT (Devlin et al., 2019). Their training approach is to "identify supported tokens in the response, given a query and a retrieved context" (Belyi et al., 2024), in order to identify hallucinated spans. According to Es et al. (2024), there are three main challenges in the evaluation of RAG systems: the retrieval system needs to identify the relevant parts, the LLM needs to exploit that part and deliver high

quality output. Saad-Falcon et al. (2024) suggest an automated framework that evaluates RAG systems with regard to their faithfulness, context relevance and answer relevance, as Es et al. (2024). In their framework a LM is used to generate a set of question-answer pairs. Three other models are then used as judge models for the three classification tasks and lastly, prediction-powered inference is conducted with a small set of human-annotated answers, to get a confidence score for the classification outputs (Saad-Falcon et al., 2024). Gao et al. (2023) train a language model to generate questions that need to be researched in order to verify the content of a given text passage. The output of the model is an attribution report that contains retrieved snippets of evidence for the content of the given text passage. Additionally, the given text passage is corrected, given the evidence, while as much information and style from the original input passage are preserved Gao et al. (2023).
Lastly, Feldman et al. (2024) in their work of evaluating RAG systems for faithfulness, find that while RAG does help to reduce hallucinations in LLMs it does not provide full protection against hallucinations.
Shuster et al. (2021) investigate the impact of RAG on dialogue settings: instead of classic QA they examine how hallucinations can be reduced in open-domain conversations, where knowledge needs to be retrieved from a database (Shuster et al., 2021).
Rau et al. (2024) deal with the problem of long context: while RAG helps with reducing hallucinations the additional context makes the processing time of the model longer. They suggest to compress the context into context embeddings Rau et al. (2024). Zhang et al. (2023a) deal with the problem of retrieval in RAG. They address the different restrictions of LLMs, namely knowledge boundaries, memory boundaries and capability boundaries, by developing a retriever that combines knowledge retrieval, memory retrieval, tool retrieval and example retrieval Zhang et al. (2023a).

One of the main parts of RAG is the context retrieval. Some of the presented approaches for RAG already include the retrieval method, for example Es et al. (2024). One independent, basic, versatile and simple retrieval method is tf-idf retrieval. This method will be introduced in the next section.

3.5 Tf-idf retrieval

This section explains a retrieval method based on term frequency inverted document frequency (tf-idf). It is based on Manning et al. (2008).
The goal of tf-idf retrieval is to retrieve documents that are semantically similar to the query. Therefore the query and the documents are embedded into a vector space. With the tf-idf retrieval a weight is computed for every term of a query-document pair. The weights are used for computing the cosine similarity between a query and a document to find the most similar ones.
The term frequency $tf_{t,d}$ computes the relative frequency of a term in a text span. It is defined as the number of times that term t occurs in document d. To avoid small numbers and zeros, the term is smoothed. The formula can be seen in 1. To get a term frequency score for a query-document pair, the sum over the tf weights for every term in both, the query and the document, is computed. It is 0 if none of the terms in the query is in the document.

$$w_{t,d} \begin{cases} 1 + log_{10}(tf_{t,d}) & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The shortcoming of term frequency alone is that frequent terms are not necessarily informative. Terms as articles, modal verbs, prepositions, ... occur very often in all of the text, so when there is a high score for a term as "the" in a query-document pair, this score does not say anything about whether the query and the document are a good match regarding their similarity.
Therefore, in addition to the term frequency, there is the inverted document frequency, idf. The goal is to get high weights for rare terms and low weights for frequent terms. This means that when a rare term such as "Time-Memory" occurs in both the query and the document in question, this should get a higher weight and contribution with regard of the similarity between the pair in question than a frequent term such as "the". The idf weight of a term is computed as the number of documents in the database divided by the number of documents in which the term occurs, as can be seen in 2. The idf weight measures the informativeness of the term.

$$idf_t = log_{10}(\frac{N}{df_t}) \tag{2}$$

To consider both the term frequency and the inverted document frequency for a term the tf-idf weight is computed, see 3. The tf-idf weight increases with the number of occurrences of a term within a document, and with the rarity of the term in the whole database.
Each term gets a tf-idf weight which is then used for computing the cosine similarity between a query-document pair.

$$w_t, d = (1 + log(tf_t, d)) \cdot log(\frac{N}{df_t}) \tag{3}$$

For cosine similarity, the query and all paragraphs/documents from the database are mapped into a vector space together. They are represented as vectors of their tf-idf weights, where the terms are the axes and the documents and query are vectors. 4 shows the formula to calculate the cosine similarity, where q_i is the tf-idf weight of term i in the query, and d_i is the tf-idf weight of term i in the document.

$$\cos(\vec{q}, \vec{d}) = \text{sim}(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{\|\vec{q}\| \|\vec{d}\|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2} \cdot \sqrt{\sum_{i=1}^{|V|} d_i^2}} \tag{4}$$

3.6 Evaluation metrics

While the detection and mitigation of hallucinations is important for the reliable deployment of LLMs in critical environments, evaluating LLM output for hallucinations is not trivial. The current section describes different evaluation methods.

RAGAs: Answer Relevance, Faithfulness, Context Relevance RAGAs (Es et al., 2024) includes three different metrics: answer relevance, faithfulness and context relevance. **Answer relevance** refers to the extent to which the generated response fits to the specific question posed. An answer is considered relevant if it clearly and appropriately responds to the question. Responses that are incomplete or include unnecessary information are penalized. It is important to note that this metric does not account for the factual correctness of the answer (Es et al., 2024). The answer relevance is computed as:

$$AR = \frac{1}{n} \sum_{i=1}^{n} sim(q, q_i) \tag{5}$$

For every response it computes the cosine similarity between the embeddings of the response and the question (Es et al., 2024).

The second metric, **faithfulness** "refers to the idea that the answer should be grounded in the given context" (Es et al., 2024). "The answer is faithful to the context, if the claims that are made in the answer can be inferred from the context" (**?**). It is computed with the help of a verification function that is contained inside a prompt (**?**). It is computed as:

$$F = \frac{|V|}{|S|} \tag{6}$$

V is the number of claims in the response that are supported by the context, and S is the total number of claims in the response (Es et al., 2024).

Context Relevance is the third of the metrics from RAGAs. It "refers to the idea that the retrieved context should be focused, containing as little irrelevant information as possible" (Es et al., 2024). A high score for context relevance means that the context "exclusively contains information that is needed to answer the question" (Es et al., 2024). This means that the context relevance is independent of the response of the model: it evaluates the context with regard to the question, and is therefore

also a measurement for the functionality and effectiveness of the retrieval method. The context relevance is computed as follows. "Given a question and its context, LLM extracts subset of sentences from the context, that are crucial to answer the question" (Es et al., 2024). Redundant information is penalized (Es et al., 2024). Context relevance is computed as:

$$CR = \frac{\text{number of extracted sentences}}{\text{total number of extracted sentences in c(q)}} \tag{7}$$

BERTscore BERTscore (Zhang et al., 2019) needs references, other than RAGAs and doesn't consider the context. It computes the cosine similiarity between terms, using the contextual embeddings of the response and the reference (Zhang et al., 2019). It is important to note, that the vector representation of a word is build based on its context, therefore a word does not always have the same representation (Zhang et al., 2019). For computing the cosine similarity a token is considered in isolation, but the surrounding tokens are represented as the contextual embedding. The following formula computes the cosine similarity between a reference token x_i and a response token $\hat{x}_j$ (Zhang et al., 2019):

$$\frac{(x_i)^T \hat{x}_j}{||x_i|| ||\hat{x}_j||} \tag{8}$$

The formula for BERTscore precision, BERTscore recall and BERTscore F-score can be found in 9, 10 and 11 respectively:

$$P = \frac{1}{|x|} \sum_{x_i \in x} max_{\hat{x}_j \in \hat{x}} (x_i)^T \hat{x}_j \tag{9}$$

$$R = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} max_{x_i \in x} (x_i)^T \hat{x}_j \tag{10}$$

$$F1 = 2\frac{PR}{P+R} \tag{11}$$

BLEUscore BLEUscore (Papineni et al., 2002) was originally developed for the evaluation of translation. It computes n-gram overlap between the source sentence and the target sentence aka translation. This means that it computes how many consecutive terms in the source sentence can be found in the target sentence (Papineni et al., 2002). This can also be applied for the evaluation of LLM responses: how strong is the word overlap between the response and the reference? While BLEUscore is not ideal for detecting hallucinations, it is a simple metric that can give insights, and is for example mentioned by Zhao et al. (2023).

3.7 Research Questions

As got clear in this section LLMs are deployed in various critical domains, including cybersecurity. They are often used as QA system and have a tendency to give vague, incorrect, ambiguous or wrong information in their responses. RAG is a well known method to reduce hallucinations of LLMs in QA scenarios. This method should be applied for the cybersecurity domain in the current work. In order to examine the impact of RAG on the performance of LLMs in a cybersecurity QA scenario, the following research questions are posed for the current work:

1. How well do LLMs perform in answering cybersecurity questions?
2. Do LLMs work better in answering questions with additional context?
3. Does the additional context have an impact on QA without answer options?
4. How do the context features impact the results?

4 RQ1: How well do LLMs perform in answering cybersecurity questions?

The first research question aims to replicate the results of Tihanyi et al. (2024) in order to examine how well LLMs perform in answering cybersecurity questions. The details of the methodology and the experiment will be illustrated in the following.

4.1 Methdology: Cybermetric (Tihanyi et al., 2024)

4.1.1 Description of question dataset

The questions used in the current work for the QA setup are taken from Tihanyi et al. (2024). Tihanyi et al. (2024) build a dataset for evaluating LLMs with regard to their knowledge about cybersecurity. Each question consists of a question, four answer options including the index A), B), C) or D) and the correct solution, containing only the corresponding correct index letter. An example can be seen in Figure 1. In the remaining work "answers" will refer to the four answer options, whereas "responses" will refer to the model output of the QA. The 10.000 questions and corresponding four multiple choice answers in Tihanyi et al. (2024) are created by feeding different resources of cybersecurity knowledge to GPT-3.5-turbo and instructing it to create the questions and answers. In various different steps, the questions are then reviewed by LLMs, post-processed and randomly reviewed by humans. From the 10.000 questions different subsets are formed, containing 80, 500, 2000 and all 10.000 questions (Tihanyi et al., 2024).
The current work uses the 2000-questions-set. Opposed to the 80- and 500-set the answers are not reviewed by experts (Tihanyi et al., 2024). Nevertheless, the 2000-set contains more questions and therefore gives more stable results. The set containing all 10.000 questions was discarded due to practical reasons: the runtime and cost for the proceeding experiments would have been too high for the scope of the current work.
The answers can be fairly long as in the example in Figure 1 or consisting only of

one or two terms.

What is an essential aspect of implementing HIPAA privacy regulations?

A: Ensuring physical security of patient records
B: Certifying all employees in security protocols
C: Ensure full legal compliance and measure implementation effectiveness with appropriate metrics
D: Developing robust encryption methods for data protection

solution: C

Figure 1: Example question of the 2000 questions dataset from Cybermetric (Tihanyi et al., 2024)

4.1.2 Description of models

Tihanyi et al. (2024) evaluate their dataset on 25 state-of-the-art models. For the current work three of them are picked: meta.llama3-1-8b-instruct-v1:0, mistral.mixtral-8x7b-instruct-v0:1 and mistral.mistral-7b-instruct-v0:2. For the rest of the current work these three models will be referred to as llama, mixtral and mistral respectively. The models are used via the AWS bedrock service. According to AWS (2024) llama has the "ability to follow instructions and tasks" and, compared to meta.llama3-8b-instruct-v1:0 has an "improved reasoning and understanding of nuances and context and multilingual translation" (AWS, 2024). Its context length amounts to 128k tokens. Mixtral is suitable for "complex reasoning analysis, text generation and code generation" (AWS, 2024) and has a maximum context length of 32k tokens (AWS, 2024). Mistral is specialized for "classification, text generation and code generation" (AWS, 2024), and has a maximum context length of 32k tokens (AWS, 2024).

4.1.3 Experimental setup

The experiments are set up according to Tihanyi et al. (2024) for ideal comparison conditions. The three models are accessed via an AWS API. The temperature for

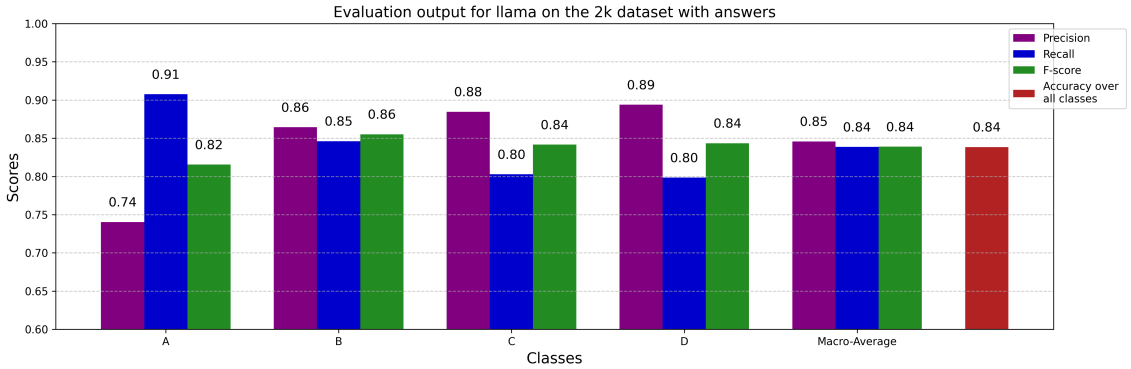
all models is set to 1.0, and the maximum generation length to 1024. The 2000 questions and their answers are prompted to the corresponding model, one question after another. The model is prompted with the instruction *You are a security expert who answers questions*, the question with its answers and the instruction *Choose the correct answer (A, B, C, or D) only. Always return in this format: 'ANSWER: X'. Keep your response as short as possible.* If the model fails to respond to a question, it retries to do so five times. In between the attempts, there is a sleeper set to $2 * *attempt$. The response of the model for each question is saved in a file for further evaluation.

4.1.4 Evaluation metric

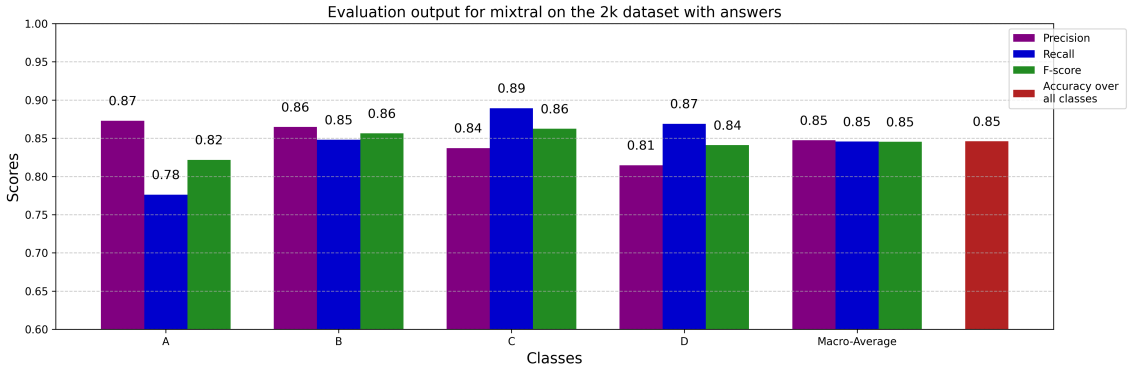
For the evaluation the index letter A), B), C) or D) is extracted from the model response and compared to the correct index letter. Then the precision, recall and f-score are computed for every class and every model, as well as a macro average precision, macro-average recall, macro-average f-score and the overall accuracy.
Additionally, an analysis with BERTopic (Grootendorst, 2022) will be performed on the whole dataset and the incorrectly answered questions. BERTopic uses a variation of tf-idf to create topic presentations. These presentations are based on clusters of document embeddings, created with the help of SBERT (Grootendorst, 2022).

4.2 Results

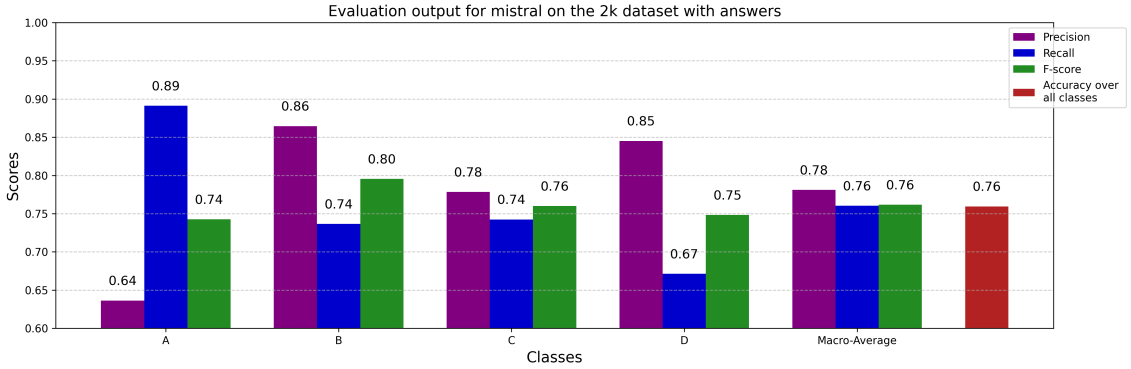
The results for prompting the 2k dataset in the described setup can be seen in Figure 2. The figure shows, for each model respectively, the precision, recall and f-score for each class, as well as the macro-average precision, macro-average recall and macro-average f-score, and the overall accuracy. In the following first the results of llama, then those of mixtral and then those of mistral will be described.
Llama reaches a macro-average precision of 0.85, a macro-average recall of 0.84 and a macro-average f-score of 0.84. The accuracy is 0.84. The lowest value is the precision for class A with 0.74 and the highest value is the recall for class A with 0.91.



(a) Performance of llama



(b) Performance of mixtral



(c) Performance of mistral

Figure 2: F-score evaluation on setup without context and answers across models: purple bars show precision per class, blue bars recall per class, green bars f-score per class, and the red bar the overall accuracy

It is important to note that the classes do not carry any meaning. The comparably low number for precision might be due to a lot of false positive picked "As" in the scenario, where the model just picks the first available answer. The same pattern can be found for the model mistral. In general, the recall reaches the lowest value for every class compared to the other metrics, except from class A. The accuracy for llama of 0.84 is higher than the accuracy for llama given by Tihanyi et al. (2024). They reached an accuracy of 0.731 on the 2k dataset.
For the model mixtral the macro-average precision, recall and f-score and the accuracy are 0.85. The lowest value is recall for class A with 0.78 and the highest value is recall for class C. The accuracy of 0.85 is lower than the accuracy of mixtral on the 2k dataset of 0.911 given by Tihanyi et al. (2024).
Lastly, the model mistral performs the worst compared to the other two models discussed. It reaches a macro-average precision of 0.78, a macro-average recall of 0.76, a macro-average f-score of 0.76 and an accuracy of 0.76. The lowest value is the precision for class A with 0.64, and the highest value is the recall for class A with 0.89. This pattern is the same as for the model llama. The accuracy is the same as the accuracy of 0.764 of mistral on the 2k dataset yielded by Tihanyi et al. (2024). Although the model is instructed to give its answer in the format "Answer: X", the mixtral and mistral partially include more in their response. Since the index letter is extracted for comparison with the solution, the evaluation results are still high, but regarding the precise compliance with the prompt instruction the model does not perform well. This is interesting when it comes to hallucinations, since 1) the model does not understand the prompt instruction well and 2) longer responses pose the risk of wrong, ambiguous, or vague information.

4.3 Analysis with BERTopic

In order to understand the nature of the incorrectly answered questions better, the data is clustered for topics with BERTopic (Grootendorst, 2022). It is important to note that the clustering was done twice, yielding different cluster topics. This means

number	all data	llama	mixtral	mistral	name
0	0.064	0.093	0.088	0.089	0_security_organization_and_standard
1	0.044	0.037	0.032	0.042	1_authentication_password_biometric_passwords
5	0.025	0.015	0.019	0.017	5_privacy_unauthorized_information_term
6	0.022	0.034	0.032	0.029	6_intrusion_detection_ids_activity
7	0.021	0.012	0.016	0.017	7_firewall_firewalls_rule_network
8	0.021	0.034	0.029	0.027	8_forensics_evidence_forensic_computer
9	0.021	0.015	0.023	0.027	9_scan_sniffing_tool_network
10	0.018	0.012	0.013	0.025	10_wireless_network_communication_technology
11	0.018	0.019	0.016	0.027	11_key_encryption_decryption_both
13	0.017	0.009	0.003	0.006	13_encryption_cryptography_main_purpose
14	0.015	0.031	0.023	0.023	14_cloud_computing_environment_without
15	0.015	0.012	0.023	0.01	15_public_key_certificate_pki
16	0.014	0.009	0.023	0.012	16_social_engineering_attacks_target
17	0.013	0.022	0.006	0.012	17_phishing_click_cybersecurity_fraud
18	0.013	0.003	0.00	0.006	18_audit_monitoring_cybersecurity_purpose
19	0.012	0.012	0.016	0.008	19_attack_machine_dos_attacker
20	0.011	0.00	0.013	0.008	20_dns_domain_record_spoofing
21	0.011	0.003	0.013	0.01	21_hash_digital_hashing_message
25	0.009	0.022	0.00	0.012	25_transposition_aes_columnar_linear
26	0.009	0.015	0.006	0.01	26_unauthorized_measure_prevent_network
29	0.008	0.00	0.003	0.006	29_vpn_virtual_private_vpns
30	0.007	0.00	0.003	0.008	30_software_source_updates_closed
31	0.007	0.006	0.003	0.002	31_database_relational_sql_transactions
36	0.006	0.003	0.01	0.002	36_hipaa_medical_privacy_healthcare
38	0.006	0.00	0.003	0.002	38_steganography_file_watermarking_overt
40	0.006	0.006	0.003	0.01	40_configuration_management_change_ccb
41	0.005	0.003	0.00	0.00	41_penetration_test_testing_reconnaissance
44	0.005	0.012	0.006	0.01	44_memory_overflow_timememory_buffer

Table 1: Most important BERtopic clusters for 2k dataset and incorrectly answered questions for each model

that BERTopic does not cluster the same data the exact same way every time, and therefore slightly different findings can be made, if the experiment is redone.
Table 1 shows the relative frequency of topics in the 2k dataset in the left column, and the relative frequency of topics in the incorrectly answered questions for each model respectively. Due to space design only the most interesting topics for the three models compared to the relative frequencies in the whole 2k dataset is shown. Tables containing all cluster topics can be found in the appendix.
Altogether, there are 44 different clusters, whereby the first cluster, topic-1, contains the outliers and can therefore be discarded.

First, the topic clusters for the 2k dataset will be described. When discarding the topic cluster for outliers, for the 2k dataset the most common topic, with a relative frequency of 0.064, is topic0 *0_security_organization_and_standard*. This means that the cluster is mainly represented by the terms *security*, *organization* and *standards*. An example question for this topic is *Who is responsible for ensuring data integrity and security for an organization?* or *What is the primary purpose of a security policy?*. The next important cluster is topic1, with a relative frequency of 0.044, named *1_authentication_password_biometric_passwords*. An example question for this topic cluster is *What is the primary purpose of biometric authentication?*. Other important cluster topics are topic2 *2_access_control_least_privilege*, topic3 *3_risk_assessment_you_management* and topic4 *4_malware_virus_malicious_type*. For the 2k dataset the relative frequency for the topics below topic23 gets very low. When looking at the topic clusters that were created for the incorrectly answered questions for llama, the following can be seen. Topic6, topic14 and topic25 have a much higher relative frequency compared to the 2k dataset, where for topic14 it is double as high, and for topic25 more than double as high. Topic44 also has more than twice the relative frequency for llama compared to the 2k dataset. The same is true for topic8, topic17 and topic26. On the other hand there are several topics where the relative frequency for the incorrectly answered questions is lower than for the whole dataset, for example topic5, topic7, topic9, topic10, topic13. For topic18 and topic21 the frequency in the incorrectly answered question set is only about a forth of the frequency of the 2k dataset. For topic20, topic29 and topic30 the fre-

quency for the incorrectly answered questions is even 0.
Comparison with the questions that were answered incorrectly by mixtral reveals the following. Topic clusters for which the relative frequency within the incorrectly answered questions is higher than for the whole dataset, are: topic0, topic6, topic8, topic14, topic15, topic16, topic40, topic44.
Topic clusters where the frequency within the incorrectly answered questions is (much) lower, are: topic1, topic13, topic17, topic18, topic21, topic25, topic31, topic36.
For mistral the data looks the following. Some of the cluster topics, where the relative frequency for the incorrectly answered questions are higher than for the whole dataset are: topic0, topic11, topic14, topic40 and topic44. There are more, but these are the ones with the highest differences. For the ones, where the frequency for the incorrectly answered questions is lower than for the whole dataset, the most important ones are: topic13, topic18, topic19, topic31, topic36, topic38, topic41.
When compared across model performance, some topics seem to be especially difficult or easy to answer. Cluster topics that have a higher frequency for the incorrectly answered questions than for the whole dataset are more difficult to answer by the models. Across models the most important ones are the topics:
0_security_organization_and_standard, *14_cloud_computing_environment_without*, *44_memory_overflow_timememory_buffer*. Cluster topics that have a lower frequency for the incorrectly answered questions than for the whole dataset, are comparably easier to answer by the models. Across models the most important ones are the topics: *13_encryption_cryptography_main_purpose* and *18_audit_monitoring_cybersecurity_purpose*.
Topic *25_transposition_aes_columnar_linear* poses an interesting issue: for llama it is especially difficult to answer, where as for mistral its frequency within the incorrect answers is only slightly higher than for the whole dataset, and for mixtral, its frequency within the incorrectly answered question is even 0.

4.4 Distinctive terms

The analysis with BERTopic gives insights on why some questions are harder to respond to by the model. The following section looks at distinctive terms that are more likely to be contained in incorrectly answered questions. According to the clusters, which are created on the basis of distinctive terms, questions containing the following words are more likely to be answered incorrectly.
For llama distinctive terms derived from the clusters are: *intrusion, detection, cloud/environment, computing, forensic(s), evidence, cybersecurity, fraud, unauthorized, memory, time-memory.*
For mixtral distinctive terms derived from the clusters are: *security, organization, intrusion, detection, forensic, evidence, cloud, environment, computing, public/key, social/engineering, attack(s), target, configuration/management, memory, time-memory.*
For mistral distinctive words derived from the clusters are: *security, organization, standards, cloud/environment. configuration/management, memory, time-memory.*
The / indicates that the terms can occur isolated or as a phrase.
While not all of these distinctive terms are contained exclusively in incorrectly answered questions, but only to a higher degree, one of them is contained in only two questions, which are always answered incorrectly by all models:
What is a primary challenge when implementing a ***Time-Memory*** *Trade-Off?*
In a ***Time-Memory*** *Trade-Off, what is the 'time' aspect referring to?*
There is no obvious reason why this is the case.
Regarding the analysis with BERTopic and the distinctive terms it should be noted that when re-running the experiments, it is not always the exact same questions that are answered incorrectly.
Another error that occurs is that of null responses, where the model does not give a response at all. For the current setup this only occurs for mistral with 7 null responses out of 481 incorrectly answered questions. Two of them fall into cluster25. Apart from that there is no pattern detectable. LLama and mixtral do not produce null responses in the current setup.

4.5 Discussion

In summary it can be said that the results yielded in this work are similar to the ones by Tihanyi et al. (2024), with some deviation. The three chosen models perform well on answering questions from the cybersecurity domain, but there are questions that are hard to respond to by the models. The approach to extract the index letter for computing the performance works, but has its shortcoming regarding the precise following of the prompt instructions. There are different possible reasons for the higher difficulty of certain cluster topics. First, the question in one cluster might be constructed in a certain way that makes it difficult for the model to understand and therefore answer it. Ambiguous or vague terms within the question can also be a reason for less understanding. Another reason can be the answers, that might be misleading, ambiguous or vaguely constructed, or too long or too short. The complexity of certain topics can also be a reason. One way to investigate this is to augment the model with additional domain specific knowledge, using RAG. The goal is to examine whether specific additional domain information will help with answering questions from specific topic clusters.

5 RQ2: Do LLMs work better in answering questions with additional context?

To investigate whether additional domain specific knowledge helps with QA in the cybersecurity domain, especially on more difficult questions, different steps need to be conducted. First, a database for context retrieval needs to be built, secondly the context needs to be retrieved, and thirdly the questions need to be augmented and prompted to the models. These steps, their experimental setup and the results will be described in detail in the following sections in order to answer the Research Question.

5.1 Methodology: Building RAG

5.1.1 Databases for retrieval

Two different databases are built for the retrieval of context, one based on NIST reports (National Institute for Standard Technology) and one based on MITRE ATT&CK . Both of them provide different context regarding cybersecurity: NIST contains, among other things, information about the scientific developments regarding cybersecurity. MITRE ATT&CK is a collection of real cyberattacks, defenses and mitigation strategies. For the context retrieval the two databases are used independently of one another. Since they contain different information in different style, a comparison of the model performance shows which of them is more suitable as context retrieval base regarding cybersecurity. The two knowledge resources are described in more detail in the following.

NIST The first database is built on NIST reports (National Institute for Standard Technology), an institute that belongs to the US department of commerce. The Federal Laboratory (Interagency) Technology Transfer Summary Reports are reports towards the president about how new technologies and scientific achievements are transferred to companies and used in economy (National Institute of Standards and Technology, 2022). The 16 reports in pdf-format are between 45 and 176 pages long. For using them as a database they are chunked into paragraphs. These paragraphs are saved as strings of text in a list, with an average of 161 characters per string, which is saved to a json file.

MITRE ATT&CK The second database is built on MITRE ATT&CK . MITRE ATT&CK is an open source knowledge base that contains information about real cyberattacks and the corresponding mitigation and defense strategies. Based on that the knowledge base contains offense and defense techniques for the development of cybersecurity in companies, science and government (Strom et al., 2018). The data is available on Github in STIX format, which is "used to exchange cyber threat intelligence" (MITRE, 2024). The knowledge base is structured into different types

of cybersecurity attacks. Each of them contains several different specific attacks. An example entry can be seen in Figure 3.

For using this data as database for retrieval all attacks are saved into one json file. During retrieval only the parts of the entry that are suitable for context are extracted: type, name and description, as will be described in more detail in Section 5.1.2.

5.1.2 Experimental setup: augmenting questions and prompting

The main part of the current experiment is to augment the questions with context. Therefore tf-idf retrieval is used as described in Section 3. Different variations of context are used for the investigation of the impact of context. The different context setups can be seen in Table 2.

name	database	length	method
N_sh_1	NIST	21 entries	top 21 entries
N_sh_2	NIST	21 entries	top 7 entries, their pre- and proceeding entry
N_sh_3	NIST	21 entries	top 3 entries, their three pre- and proceeding entries
N_lo_1	NIST	210 entries	top 210 entries
N_lo_2	NIST	210 entries	top 70 entries, their pre- and proceeding entry
N_lo_3	NIST	210 entries	top 30 entries, their three pre- and proceeding entries
M_sh_d	MITRE	6 entries	top 6 descriptions
M_sh_dt	MITRE	6 entries	top 3 descriptions, the corresponding type
M_sh_dn	MITRE	6 entries	top 3 descriptions, the corresponding name
M_sh_dtn	MITRE	6 entries	top 2 descriptions, the corresponding type and name
M_lo_d	MITRE	36 entries	top 36 descriptions
M_lo_dt	MITRE	36 entries	top 18 descriptions, the corresponding type
M_lo_dn	MITRE	36 entries	top 18 descriptions, the corresponding name
M_lo_dtn	MITRE	36 entries	top 12 descriptions, the corresponding type and name

Table 2: Context setups for RAG

```
{
    "type": "bundle",
    "id": "bundle--dc56d497-eed2-4a12-9584-b6fb95b39430",
    "spec_version": "2.0",
    "objects": [
        {
            "x_mitre_domains": ["enterprise-attack"],
            "object_marking_refs": ["marking-definition--fa42a846-8
               d90-4e51-bc29-71d5b4802168"],
            "id": "course-of-action--0bc3ce00-83bc-4a92-a042-79ffbc
               6af259",
            "type": "course-of-action",
            "created": "2018-10-17T00:14:20.652Z",
            "created_by_ref": "identity--c78cb6e5-0c4b-4611-8297-d1
               b8b55e40b5",
            "external_references": [
                {
                    "source_name": "mitre-attack",
                    "url": "https://attack.mitre.org/mitigations/T1
                       084",
                    "external_id": "T1084"
                },
                {
                    "source_name": "FireEye WMI 2015",
                    "description": "Ballenthin, W., et al. (2015)
                       ...",
                    "url": "https://www.fireeye.com/.../wp-windows-
                       management-instrumentation.pdf"
                }
            ],
            "modified": "2019-07-25T12:35:09.565Z",
            "name": "Windows Management Instrumentation Event
               Subscription Mitigation",
            "description": "Disabling WMI services may cause system
                instability...",
            "x_mitre_deprecated": true,
            "x_mitre_version": "1.0",
            "x_mitre_modified_by_ref": "identity--c78cb6e5-0c4b-461
               1-8297-d1b8b55e40b5"
        }
    ]
}
```


Figure 3: Example attack in enterprise attacks:
course-of-action–0bc3ce00-83bc-4a92-a042-79ffbc6af259 (MITRE, 2024)

For the NIST database the context is retrieved in three different ways. For the first one, the x closest paragraphs are retrieved. For the second one, the x/3 closest ones are retrieved, and the one preceding and proceeding them. For the third one the x/7 closest ones are retrieved and the three preceding and proceeding them.
A second parameter for the retrieval is the context length: to examine the impact of the context length, the models are prompted with the shortest possible context of x=21. This is the shortest possible context, because for every setup it should be of the same length: 3x7=21. The length of the long context was determined based on the maximum context token length of the models. Since the context should have the same length for all models the restriction is 32k, due to the token length restriction of mixtral and mistral.
For the MITRE ATT&CK database four different combinations of context are used: first, only the descriptions of the x highest ranked attacks are used as context. Secondly, for x/2 highest ranked descriptions, the name of the attack is retrieved additionally to the description. Thirdly, for the x/2 highest ranked descriptions the type of the attack is retrieved additionally to the description. Fourthly, for the x/3 highest ranked descriptions the type and the name are retrieved additionally to the description. Here the smallest x is set to 6, the highest x, for maximum context, to 36. Longer context than 36 instances produced too many null responses, therefore a relatively small number for the maximum length was chosen to keep the experiments feasible and the results comparable.

For the retrieval, first, the questions are pre-processed, so only nouns, verbs, adjectives and pronouns are used for computing the similarity with the entries in the databases. Then a python library is used for automatically conducting the tf-idf retrieval including creating the vector space and computing cosine similarities. Depending on the desired setup (see Table 2), the context for each question is retrieved and saved to a file together with the question.
The prompting then was carried out in the same way as for RQ1, see Section 5.1.2: one question after another is prompted to the LLM together with its context. The prompt contains the instruction *You are a security expert who answers questions*,

the question with context and its answers, and the instruction *Choose the correct answer (A, B, C, or D) only. Always return in this format: 'ANSWER: X'. Take the given context into consideration when answering the question.* The model temperature is set to 1.0, and the maximum generation length to 1024. If the model fails to respond to a question, it retries to do so five times. In between the attempts there is a sleeper set to $2 * *attempt$. The model responses are saved to a file for further evaluation.

5.1.3 Evaluation metrics

For the evaluation of the responses, three evaluation metrics are chosen. First, as for RQ1, see Section 4.1.4, the precision, recall, f-score and accuracy for the responses are computed based on the overlap of the index letters.
As already mentioned in Section 4 the responses of mixtral and mistral are longer and more extended than just *ANSWER: X*. While this indicates poor compliance with the prompt instruction, the responses are used for comparison with the setup in RQ3, see Section 6.2. RQ3 examines freely generated answers - without options and index letters. For comparison the extended responses of mixtral and mistral for RQ1 and RQ2 are evaluated. Therefore, RAGAs' answer relevance and faithfulness, BERTscore and BLEUscore are used, as described in Section 3.
As opposed to RAGAs BERTscore and BLEUscore require references. Therefore, the correct solution out of the answer options is taken as reference, including only the answer without the index letter.

5.2 Results

5.2.1 F-score

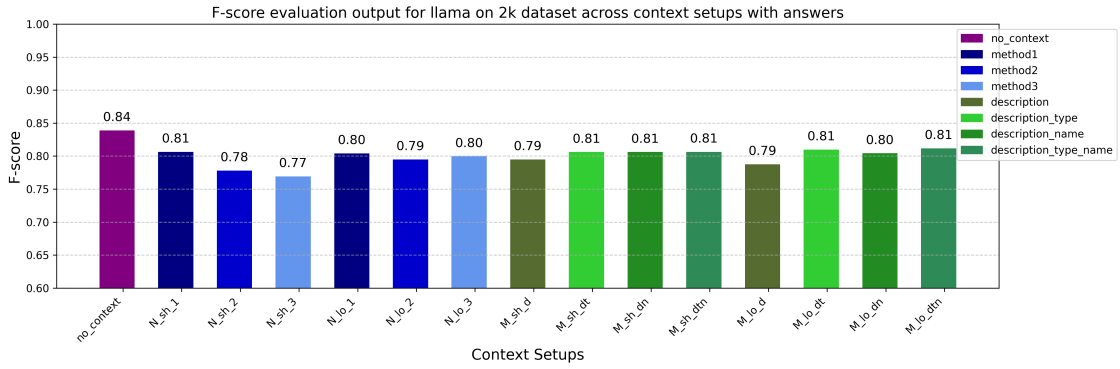
The first evaluation metric is to compute the precision, recall, f1-score and accuracy for the overlap between the chosen response and the solution, both represented by their index letters. The f1-scores for all models across setups can be found in

Figure 4. For better comparison, the first bar shows the macro f-score for the setup without context, see RQ1 (Section 4.2).

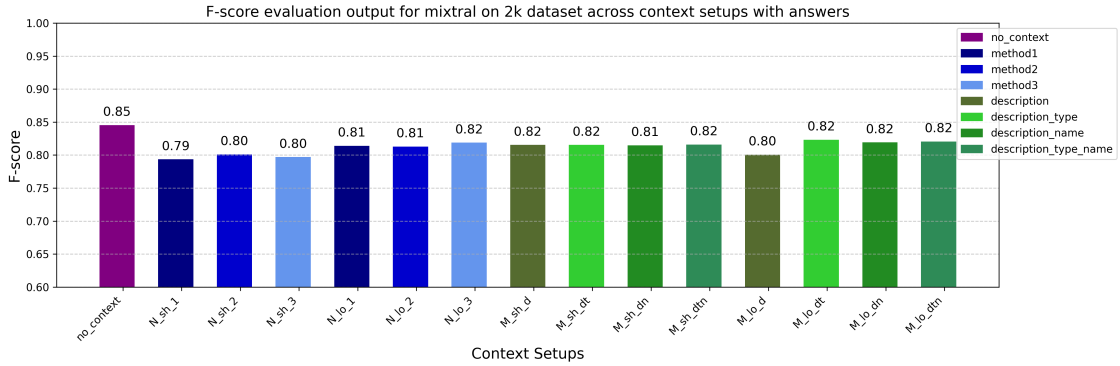
In general it can be observed that the augmentation with context does not improve the performance of the model. For all models the best f1-score is yielded by the setup without context. For llama the difference between the highest score across the context setups and the score for the setup without context is 0.03; for mixtral it is 0.83 too, and for mistral 0.04. The deviations across the context setups are comparable small within a model. In the following the performance of the models will be compared, followed by a more in depth analysis of the impact of the context length, the source of context and the type of context.

Comparison of models When comparing the models it gets clear that mistral, see Figure 4c, performs worst compared to llama and mixtral. Mistral does not outperform any of the other models in any other setup. The best performance is yielded by mixtral with an average score of 0.811 compared to 0.799 by llama. Llama has the highest deviation in peformance across the context setups, with a minimum score of 0.77 for N_sh_3 and a maximum score of 0.81 for several setups.

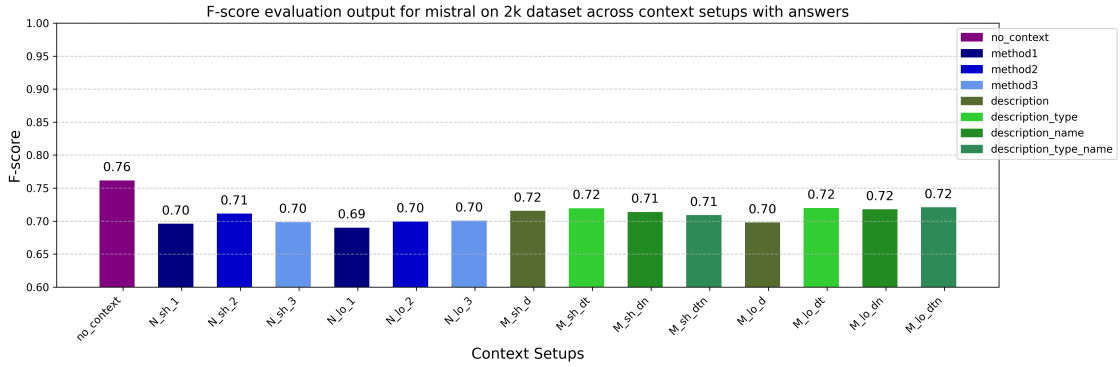
Impact of context length For llama, see Figure 4a, on average the longer context based on NIST yields better results than the short context based on NIST. The average score for long context setups based on NIST is 0.797, compared to 0.787 for short NIST context setups. For MITRE ATT&CK the long context setups yield an average score of 0.8025, the short context setups an average score of 0.805. Therefore for MITRE ATT&CK the short context setups yield a slightly higher score.
For mixtral, see Figure 4b, the following is observable for the impact of the context length. For the context based on NIST the longer context yields better results, of 0.813 on average, compared to an average of 0.797 for the short context setups. For MITRE ATT&CK the short context setups yield an average score of 0.8175, compared to an average score of 0.815 for the long context setups. Therefore, as for llama, for MITRE ATT&CK the short context setups are slightly better.
For mistral, see Figure 4c, for both context based on NIST and context based on



(a) Performance of llama



(b) Performance of mixtral



(c) Performance of mistral

Figure 4: F-score evaluation across context setups with answers: the purple bar shows prompting without RAG, blue bars show the context based on NIST, green bars show the context based on MITRE ATT&CK

MITRE ATT&CK the difference between short context and long context setups on average is almost 0.
To summarize, for MITRE ATT&CK the short context seems to work slightly better, whereas for NIST the long context seems to be better. Nevertheless the differences are quite small.

Impact of source of context When looking at the performance differences between the context setups based on NIST and the context setups based on MITRE ATT&CK it gets clear, that MITRE ATT&CK yields higher scores for all models. While some context setups based on NIST outperform other context setups based on MITRE ATT&CK , on average MITRE ATT&CK performs better. For example, for mixtral, the context setup N_lo_3 with a score of 0.82 is better than the context setup for M_lo_d with a score of 0.80, but the average of all NIST setups is 0.805, while the average for all MITRE ATT&CK setups is 0.816. This is true for all models.

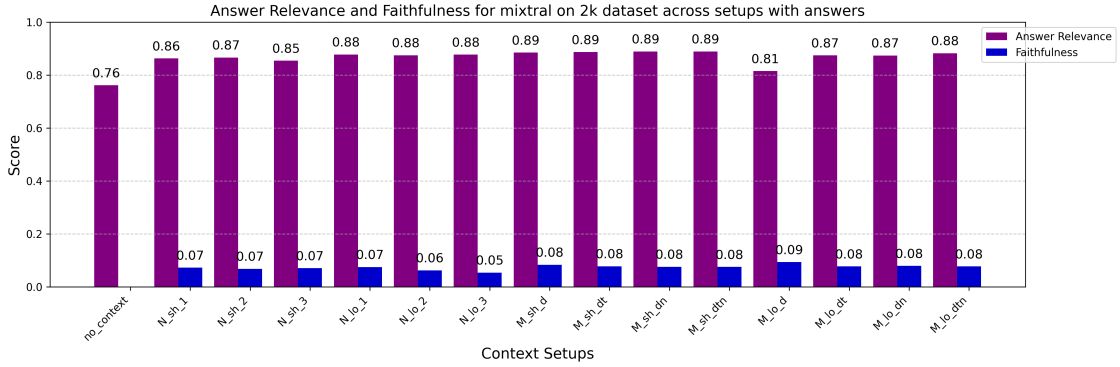
Impact of type of context Looking at the different types of context, dependent on the retrieval parameters, the following can be observed. For llama, for the NIST setups, the context retrieved with method1 yield better scores than the ones retrieved with method2 and method3, for both the short and the long context. On the other hand, for MITRE ATT&CK the context that contain only descriptions perform worse compared to the other MITRE ATT&CK context setups, for both the short and the long context.
For mixtral this pattern is not as clear. For NIST none of the types of contexts is clearly better than the others. For MITRE ATT&CK again the long context that contains only descriptions, M_lo_d , performs worse than the others, also than the short one containing only descriptions, M_sh_d .
For mistral, for long NIST the context retrieved with method2 outperforms the contexts retrieved with method1 and method3. For MITRE ATT&CK , again, the long context containing only descriptions, M_lo_d , performs worst compared to the other MITRE ATT&CK context setups.

To summarize it can be said, that the setup without context works best. The model mistral performs worst, followed by llama, and mixtral performs best. The deviations between the context setups within a model are small. The context based on MITRE ATT&CK yields better results than the context based on NIST. For NIST the long context setups work better than the short ones.

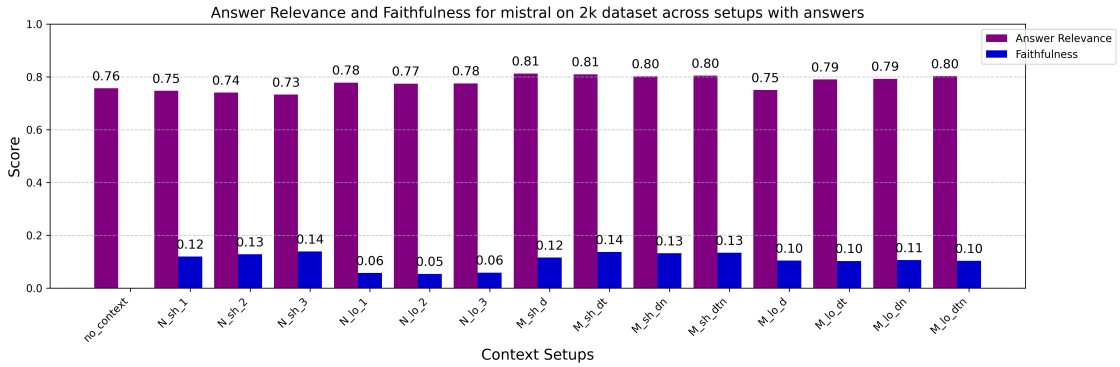
5.2.2 RAGAs

As already mentioned before the responses of mixtral and mistral contain not only the index letter but more explanations. Therefore, these responses are evaluated with RAGAs for comparison with the responses that are generated without answers in RQ3, see Section 6. For llama this is not possible, because the response contains only the index letter. Figure 5 shows the scores for answer relevance and faithfulness across setups for mixtral and mistral respectively. The answer relevance provides information about how relevant the response is to the question, while the faithfulness computes how well the response can be deducted from the context. Therefore there is no score for the faithfulness for the setup without context. For every setup and model only one value is computed for the whole dataset. The given value is the score for the answer relevance and faithfulness for the whole dataset, respectively. In general it can be seen that mixtral performs better than mistral, where no score of mistral outperforms any score of mixtral, except from the setup without context. Opposed to the evaluation on index letters, the setup without context yields the worst score for answer relevance for mixtral and mistral, with a score of 0.76 for both models. The following paragraphs will capture the Answer Relevance and Faithfulness, each, again talking about the impact of the context length, the source of context and the context type.

Answer Relevance Regarding the answer relevance the following can be observed for mixtral. For NIST the answer relevance is higher for long context setups with



(a) Performance of mixtral



(b) Performance of mistral

Figure 5: Answer Relevance and Faithfulness (RAGAs) across setups for mixtral and mistral with answers: purple bars show the answer relevance, blue bars the faithfulness for each context setup

an average score of 0.88, compared to an average score of 0.86 for the short NIST context setups. The opposite is true for MITRE ATT&CK where the short context setups yield an average score of 0.89, outperforming the long context setups with an average score of 0.858. The same can be observed for mistral. With context based on NIST the long context outperforms the short context with an average score of 0.777 compared to an average score of 0.74. With the context based on MITRE ATT&CK on the other hand the short context outperforms the long context, with an average score of 0.805 compared to an average score of 0.78.

When looking at the source of context for mixtral the context based on MITRE ATT&CK performs slightly better than the context based on NIST with a score of 0.874 compared to a score of 0.87. Nevertheless, this difference is very small. For mistral the difference is higher: context setups based on MITRE ATT&CK yield an average score of 0.794, whereas context setups based on NIST yield an average score of 0.758.

Regarding the type of retrieval it is interesting, that the long context of MITRE ATT&CK containing only descriptions, M_lo_d , again yields a lower score compared to the other context setups based on MITRE ATT&CK .

When looking at the faithfulness the following can be observed. The scores for faithfulness are in general quite low. For mixtral, on average the scores for context retrieved from MITRE ATT&CK are higher than for context retrieved from NIST, with an average of 0.081 for MITRE ATT&CK and an average of 0.065 for NIST . Interestingly, the highest score for faithfulness occurs for the setup with the lowest score for answer relevance: long context based on MITRE ATT&CK that contains only descriptions. For NIST the short context is slightly better than the long context. For mistral the scores for faithfulness are higher than for mixtral. Long context setups lead to a higher faithfulness than short context setups for both, context based on NIST and context based on MITRE ATT&CK . The average score for short context with NIST is 0.13 compared to 0.056 for long context. For MITRE ATT&CK the average score for short context is 0.13 compared to an average of 0.103 for long context. This shows that for short context NIST and MITRE ATT&CK yield the same score. With long context MITRE ATT&CK outperforms NIST. For the

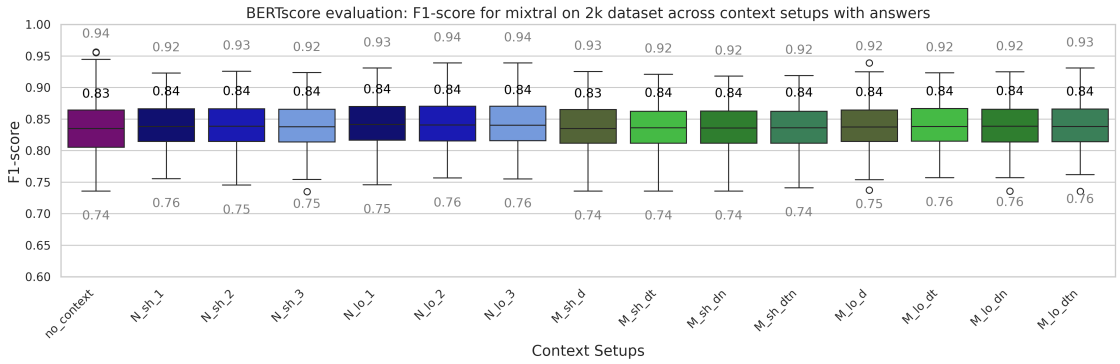
short NIST context that yields a comparably low score for answer relevance the score for faithfulness is comparably high. Regarding the faithfulness, mistral outperforms mixtral clearly. Nevertheless the scores are very low altogether.

To summarize, the answers seem to be more relevant to the question for the setups with context, compared to the setup without context, especially for mixtral. For mixtral all context setups outperform the setup without context. For mistral, the long NIST context setups and all MITRE ATT&CK context setups, except from M_lo_d , outperform the setup without context. Nevertheless, looking at the scores for faithfulness, the responses do not seem to rely on the context too much.

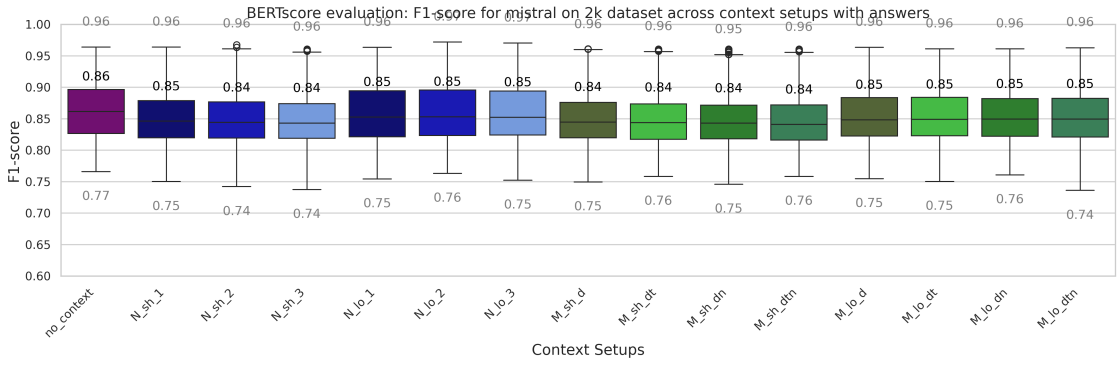
To use a simpler metric for easy comparison the responses of mixtral and mistral will be evaluated with BERTscore and BLEUscore in the following.

5.2.3 BERTscore

BERTscore measures the similarity between two text strings, in this case the similarity between the response and the reference. When looking at Figure 6 it gets clear that there is little deviation for the scores between the setups. For mixtral the median is always 0.84. There are almost no outliers. The scores for the upper whiskers range between 0.92 and 0.94, for the lower whisker between 0.74 and 0.76. For mistral the median ranges between 0.84 and 0.85, and 0.86 for long context based on MITRE ATT&CK containing descriptions, types and names. The scores for the upper whiskers range between 0.95 and 0.96, and for the lower whiskers between 0.74 and 0.76. For mistral there is more deviation within the quartiles across the setups, compared to mixtral. This indicates that on average there is a high semantic similarity between the model responses and the references, that does not differ too much across setups. Nevertheless, these results have to be taken with a grain of salt, because BERTscore usually is fine-tuned with the desired data, which was not done in the current work.

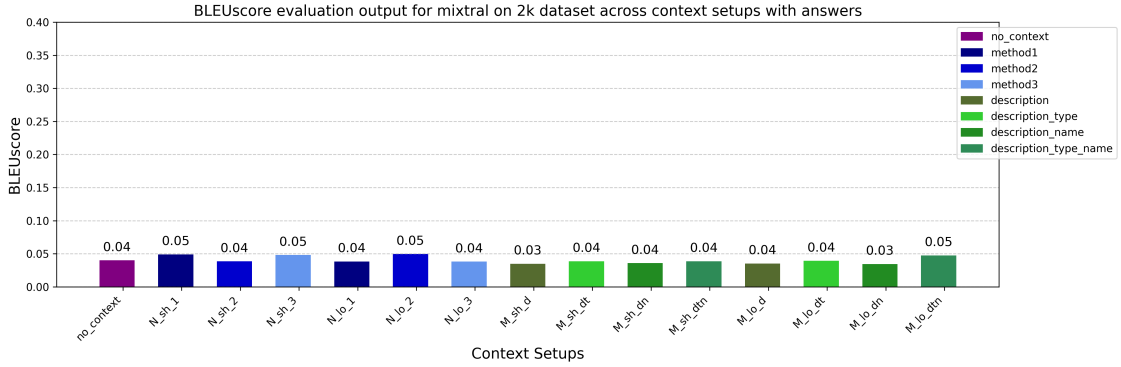


(a) Performance of mixtral

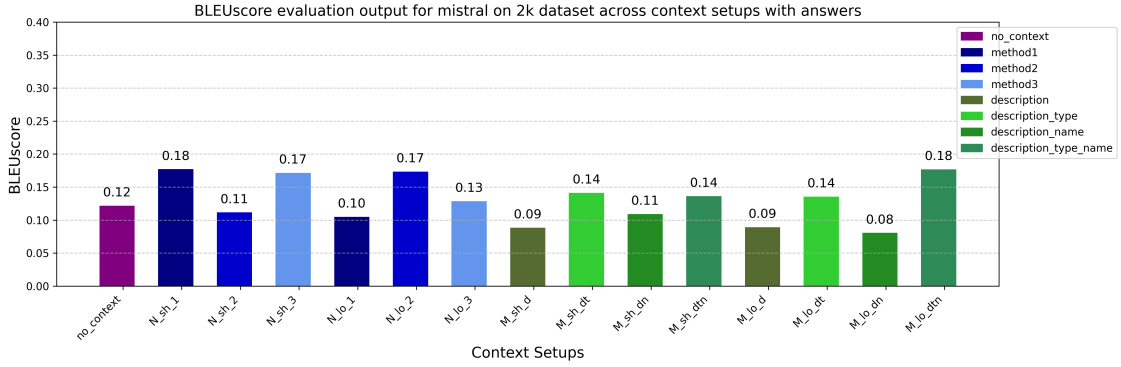


(b) Performance of mistral

Figure 6: BERTscore evaluation across setups for mixtral and mistral with options: purple box shows the result for setup without context, blue boxes for the setup with NIST, green boxes for the setup with MITRE ATT&CK



(a) Performance of mixtral



(b) Performance of mistral

Figure 7: BLEUscore evaluation across setups for mixtral and mistral with answers: purple bar shows the result for setups without context, blue bars for the setups with NIST, green bars for the setup with MITRE ATT&CK

5.2.4 BLEUscore

BLEUscore computes the n-gram overlap between two strings of texts, in this case the n-gram term overlap between the response and the reference. Investigating the results of the evaluation with BLEUscore, see Figure 7, reveals interesting things. For mixtral the scores are almost 0, ranging between 0.03 and 0.05. For the context setups based on NIST, the short context retrieved with method1 and method3 performs better than the short context retrieved with method2. For long context based on NIST context retrieved with method2 performs better than method1 and method3. Nevertheless, the values are too small to interpret the performance differences between the setups.
For mistral the scores are higher. For NIST the short context yields better results than the long context with an average score of 0.153 versus an average score of 0.133. For MITRE ATT&CK the opposite is true, where the long context setups slightly outperform the short context setups. Nevertheless the difference is quite small with an average score of 0.123 versus 0.120. AS can be seen for these scores the context based on NIST outperforms the context based on MITRE ATT&CK . For NIST they same can be observed as for mixtral: for the short context the context retrieved with method1 and method3 perform better than the context retrieved with method2, whereas for the long context the opposite is true: long context retrieved with method2 yields better results than long context retrieved with method1 and method3. For MITRE ATT&CK it can be seen from the figure that for both, the short and the long context, the context setups that contain descriptions and type and descriptions, type and name outperform the context setups that contain only descriptions or descriptions and name.
To sum it up the results show that there is very little n-gram overlap between the responses and the references. This is interesting because the correct solution out of the answers is taken as reference for each question. On the other hand, for some cases the responses are much longer than the references which also impacts the BLEUscore. Lastly, the n-gram overlap does not provide too much information about the quality of the response.

As was seen with all the evaluation metrics, there are questions that are answered incorrectly. These questions will be analysed in the following.

5.3 Error Analysis

This section takes a look at the errors made by the models for the experiment with answers using RAG. The focus should be on two aspects: first the questions, that were mostly answered incorrectly for RQ1, see section 4.2, and secondly the null responses produced by the models for RQ2, see section 5.3.

Distinctive Terms The motivation for adding context is to investigate whether the context helps with questions that are hard to answer by the models. As mentioned in section 4.4 one distinctive term, and therefore two questions, are responded to incorrectly by all models:
What is a primary challenge when implementing a ***Time-Memory*** *Trade-Off?*
In a ***Time-Memory*** *Trade-Off, what is the 'time' aspect referring to?*
When looking at the responses generated with additional knowledge the following can be observed: For most of the setups and models these two questions are still answered incorrectly. There are six cases, all answered by llama, all with context based on MITRE ATT&CK , where only the first of the two questions is responded to incorrectly. The second one is answered correctly in these scenarios.
Looking at the other terms, that occurred frequently in incorrectly answered questions for RQ1 (see section 4.4), the following is revealed. On average the context does not help with the questions containing these terms. In some cases more questions containing these terms are answered incorrectly. Considering the fact, that in general the models perform worse with context and therefore there are more incorrectly answered questions altogether, this is not surprising.

Null Responses Null responses are cases, where the model does not respond, leading to the response "null". For the setup without context out of all models only mistral produces null responses, where for the setups with context all models produce

setup	llama	mixtral	mistral	summary
no context	0/323 = 0	0/308 = 0	7/481 = 0.015	7/1112 = 0.006
N_sh_1	1/388 = 0.03	22/420 = 0.052	30/618 = 0.049	53/1426 = 0.037
N_sh_2	6/448 = 0.013	25/408 = 0.061	31/588 = 0.053	62/1444 = 0.043
N_sh_3	7/465 = 0.015	22/414 = 0.053	26/610 = 0.043	55/1489 = 0.037
N_lo_1	0/392 = 0	26/382 = 0.068	17/623 = 0.027	44/1397 = 0.031
N_lo_2	1/411 = 0.002	24/384 = 0.063	14/604 = 0.023	33/1399 = 0.028
N_lo_3	1/401 = 0.003	24/372 = 0.065	13/602 = 0.022	38/1375 = 0.028
M_sh_d	23/420 = 0.055	25/378 = 0.066	33/580 = 0.057	81/1378 = 0.059
M_sh_dt	20/397 = 0.050	18/375 = 0.048	23/568 = 0.040	61/1340 = 0.046
M_sh_dn	16/395 = 0.041	17/376 = 0.045	28/581 = 0.048	61/1352 = 0.045
M_sh_dtn	12/393 = 0.031	14/372 = 0.038	23/588 = 0.039	49/1353 = 0.036
M_lo_d	323/491 = 0.658	290/458 = 0.633	491/659 = 0.745	1104/1608 = 0.687
M_lo_dt	350/405 = 0.864	319/374 = 0.853	529/584 = 0.906	1198/1363 = 0.879
M_lo_dn	359/414 = 0.867	335/390 = 0.859	528/583 = 0.906	1222/1387 = 0.881
M_lo_dtn	358/394 = 0.909	345/381 = 0.906	534/570 = 0.937	1237/1345 = 0.920

Table 3: Frequency of null responses within incorrect responses across models and setups for setup with answers

null response. Table 3 shows the percentage of null responses within the incorrect responses across setups for every model, as well as the summary for all models. It gets clear that the highest amounts of null responses appear for the long context based on MITRE ATT&CK , the highest frequency being 0.937 for mistral on the long setup including description, type and name. The other scores for the relative frequency of null responses within these context setups range between 0.633 and 0.909 for mistral. The average for llama on these setups is 0.8245, for mixtral 0.813 and for mistral 0.873. Since the evaluation with f-score show lower scores for the setups with context than for the setup without context the question arises whether the null responses are responsible for the performance drop. For short and long context based on NIST and for short context based on MITRE ATT&CK the percentage of null responses within the incorrectly answered questions is so low, that this is most probably not the case, see Table 3. For the long context based on MITRE ATT&CK almost all incorrectly answered questions are null responses. Nevertheless the scores for these setups are comparably high, see Figure 2. One hypothesis is that the long context based on MITRE ATT&CK is too long, leading to prompting errors and therefore null responses. Another hypothesis is, that the context is too long and therefore too misleading, leading the model to exceed the maximum response time, when trying to respond to the question. It is important to note that there are not a lot more incorrect questions for long context based on MITRE ATT&CK than for the other setups, but almost all of them are null responses. Therefore it is not clear whether the model would respond to the question correctly or incorrectly if the context didn't cause prompting or time exceeding errors. It is also not clear whether these errors are caused by the structure or by the content of the context.

5.4 Discussion

Altogether, the results show that additional knowledge does not improve the performance of LLMs in a QA scenario in the cybersecurity domain. Regarding the answer relevance the results are quite well, and the setups with context outperform the setup without context. When using the full responses of mixtral and mistral for

this evaluation, instead of extracting the index letter, it is shown that the responses are relevant with regard to the question. Nevertheless, they do not rely on the context too a high degree. BERTscore results for the whole responses of mixtral and mistral show almost no deviation across the different context setups, including the setup without context. This indicates that for all setups there is a comparably high semantic similarity between the response and the reference. BLEUscore on the other hand shows very little n-gram overlap between the responses and the references. It is interesting to see that the scores for mistral are clearly higher than those for mixtral. A probable reason for this is that mistral sticks less to the prompt instruction to rely on the context, and therefore its response is more influenced by the answer options and therefore more similar to the reference which is taken from the answer options, leading to a higher n-gram overlap.

This difference in compliance with the prompt instruction can also be observed for the evaluation with f-score. There, the highest deviation across context setups is yielded by llama. This confirms the observation, that llama follows the prompt instruction more closely than mixtral and mistral. It indicates that, because llama sticks to the instruction to rely on the context for finding the correct solution, there is higher deviation in performance due to the different context, which may mislead the model with regard to the answer options. The evaluation with f-score on the index letters, and the amount of null responses for the setups with context, show clearly that the additional context does not help the LLMs with answering the options by picking the correct answer option. One reason for this could be that the context is distracting for the model, leading it to pick the wrong answer out of the four options. This is especially interesting because the answer relevance from RAGAs shows quite good results, indicating that when the model elaborates within the response instead of just picking one of the options, the relevance towards the question gets higher. On the other hand the faithfulness of RAGAs shows very poor results, indicating that the response does not rely on the context.

To further investigate this the next experimental step is to prompt the question without answer options and examine whether context helps the LLMs answering the questions when no answer options are available and freely generated responses

are required.

6 RQ3: Does the additional context have an impact on questions without answer options?

As was shown in RQ2, see section 5 the context does not help with answering questions. The performance decreases and the amount of incorrect questions and null responses increases. The context does not have a positive impact on the questions with distinctive terms that are more often responded to incorrectly. For RQ1, section 4, and RQ2, section 5, the models are prompted with the question, the answers, the instructions, and the context for RQ2. The next step is to investigate the impact of context for questions without answers. The approach will be discussed in this section.

6.1 Methodology: QA without answer options

6.1.1 Questions and models

For this research step the same questions are used as for RQ1 and RQ2, see Section 4.1. The answer options are taken away, but the 2000 questions stay the same. The experiment without answers is conducted without context and with context, to investigate the effect of adding domain specific context. The same context setups are used as for RQ2, see Section 5.1. The same models are prompted as for RQ1 and RQ2, see Section 4.1.

6.1.2 Experimental setup

Again, the models are prompted via an API. The temperature is set to 1.0. The questions are prompted one after another. For the setup without context the prompt consist of the instruction *You are a security experts who answers questions*, the

question, and the instruction *Answer the question as short and precisely as you can. Use maximum 120 tokens for your response.* For the setups with context the prompt consists of the instruction *You are a security experts who answers questions*, the question, the context, and the instruction *Answer the question as short and precisely as you can. Rely on the provided context to answer the question. Do not include phrases such as 'according to the provided context' or similar ones in your answer. Use maximum 120 tokens for your response.* Since the answer options serve as the reference, and they are not long strings of text 120 tokens as instructed restrictions is enough.
As for the other models the maximum generation length is set to 1024. Although the model is instructed to respond with only 120 tokens, the maximum generation length is set to 1024, as for RQ1 and RQ2, for better comparison and to restrict the output length, in the case that the model does not follow the prompt instruction. As for the first two research questions, if the model fails to respond to a question, it retries to do so five times. In between the attempts there is a sleeper set to $2 * *attempt$.

6.1.3 Evaluation metrics

Since for the current research step there is no index letter to be extracted and compared to the solution index letter in order to compute f-score, the evaluation relies on RAGAs' answer relevance and faithfulness, BERTscore and BLEUscore. This evaluation metrics are used in the same way as already described in Sections 3 and 5.1.3.

To not only rely on the evaluation results of RAGAs BERTscore is also used for evaluation. As already mentioned in Section 5.1 BERTscore requires references. Again, the correct solution out of the answer options serves as reference for each question. Compared to the dataset with answers three questions are excluded from the dataset because their correct solution is "none of the above given options", which can not be used as a reference answer for BERTscore. Since this case occurs only three times, the difference can be neglected for comparison.

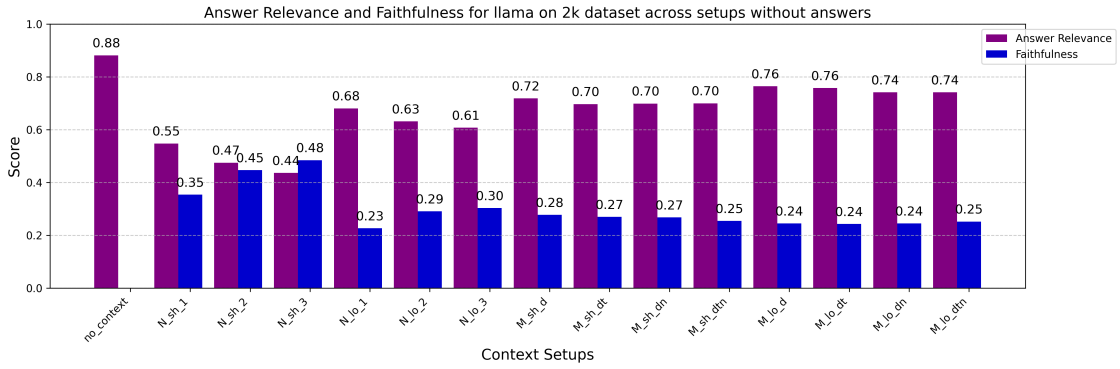
6.2 Results

This section illustrates and discusses the results for QA without answers.

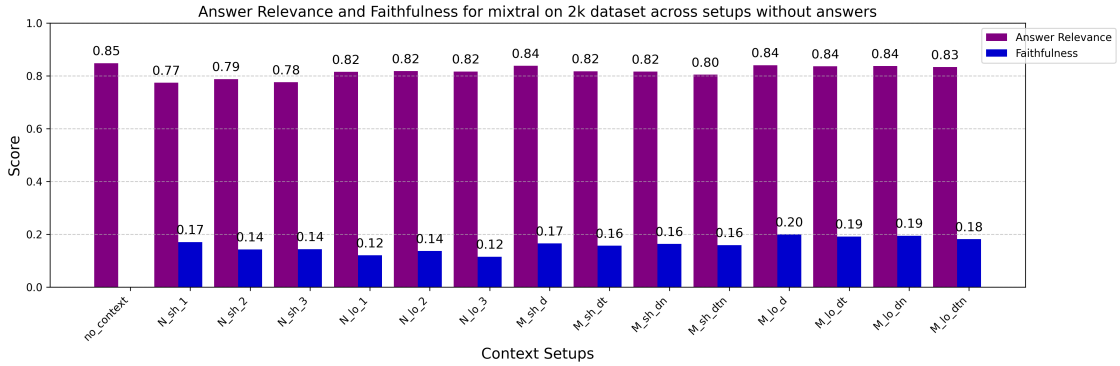
6.2.1 RAGAs

Figure 8 shows the results for the evaluation with answer relevance and faithfulness. For every setup and model only one value is computed for the whole dataset. The given value is the score for the answer relevance and faithfulness for the whole dataset, respectively. As for RQ2 first the overall model performance will be compared, followed by descriptions of the impact of context length, the source of context and the type of context.

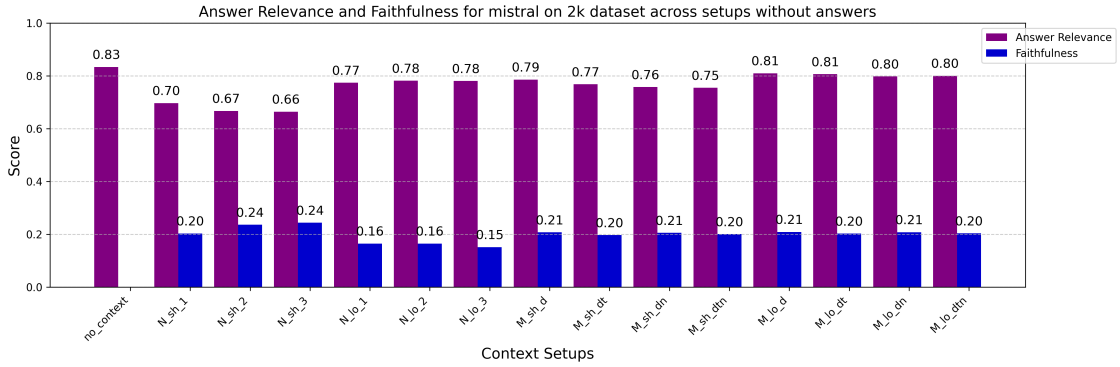
Answer Relevance For the answer relevance the following can be observed. Mixtral performs best across setups, followed by mistral. On average, llama performs worst regarding the answer relevance. For all models the setup without context yields the highest score. For the setup without context llama performs best with a score of 0.88, compared to mixtral with 0.85, and mistral with 0.83.
Starting with llama longer context leads to higher answer relevance, for both context based on NIST and context based on MITRE ATT&CK . The average score for long NIST context is 0.64, compared to an average of 0.49 for the short NIST context. For MITRE ATT&CK the average for long context is 0.75, for short context 0.705. The pattern can be observed for mixtral and mistral too, although the difference is not as severe. For mixtral the average score for long NIST context is 0.82 outperforming the short context with a score of 0.78. The score for long MITRE ATT&CK context is 0.8375, outperforming the short context with a score of 0.82. For mistral the long NIST context again outperforms the short NIST context with an average score of 0.777 versus an average score of 0.673. The same pattern is true for context based on MITRE ATT&CK . The long context yields an average score of 0.805, outperforming the short context with an average score of 0.768.
From these numbers it can already be seen that the context setups built on MITRE ATT&CK yield better results than the context setups built on NIST. For llama the



(a) Performance of llama



(b) Performance of mixtral



(c) Performance of mistral

Figure 8: Answer Relevance and Faithfulness (RAGAs) evaluation across setups across models without answers: purple bars show the answer relevance, blue bars the faithfulness for each context setup

context setups based on MITRE ATT&CK get an average score of 0.73 outperforming the context setups based on NIST with an average score of 0.56. The same is true for mixtral. Here the average score for MITRE ATT&CK is 0.829 compared to the average score for NIST of 0.79. The same pattern is observed for mistral with an average score of 0.78 for MITRE ATT&CK and an average score of 0.727 for NIST. It should be noted that the differences for llama are the highest, followed by the differences for mistral, whereas for mixtral the differences are quite low. This is true for the impact of the context length, as well as for the impact of the source of context.
It is interesting to see that the scores for short context based on NIST for llama are quite low compared to the other scores. Llama seems to be very sensitive towards the short NIST context. Apart from that nothing outstanding can be found for the type of context.

Faithfulness Moving on to faithfulness the following can be observed. On average llama performs best. This is contrasting the performance with regard to answer relevance. Mixtral performs worst with regard to the faithfulness, which also contrasts its best performance with regard to answer relevance.
Looking at the impact of the context length, the following can be observed. For llama the short context outperforms the long context, for both the setups based on NIST and the setups based on MITRE ATT&CK . The average score for short NIST context is 0.427, outperforming the long context with an average score of 0.273. The average score for short context based on MITRE ATT&CK is 0.2675 outperforming the long context with an average score of 0.2425. For mixtral the following can be observed. When looking at the context based on NIST the short context outperforms the long context with an average score of 0.15 versus an average score of 0.127. The opposite is true for context based on MITRE ATT&CK : the long context with an average score of 0.19 outperforms the short context with an average score of 0.1625. For mistral the following can be observed. For the context based on NIST the short one outperforms the long one with an average score of 0.227 versus an average score of 0.157. For the context based on MITRE ATT&CK there is no difference between

the long and the short context: both reach an average score of 0.205.
When comparing the impact of the two context resources the following can be observed. For llama the context based on NIST with a score of 0.35 clearly outperforms the one based on MITRE ATT&CK with a score of 0.255. For mixtral instead the context based on MITRE ATT&CK outperforms the context based on NIST with a score of 0.176 versus an average score of 0.138. For mistral the context based on MITRE ATT&CK very slightly outperforms the one based on NIST with an average score of 0.205 versus 0.192.
On the one hand the scores for faithfulness are quite low, which indicates that the context is not helpful for answering the questions. On the other hand, the scores are higher than for the setup with answer options. This suggests that without answer options the model relies more on the context for responding to the question. Nevertheless, this observation could also be due to the deviation in the prompt length compared to the setup with answer options. There is also a relation between comparably high scores for faithfulness and comparably low scores for answer relevance. This is especially visible for the performance of llama with short context based on NIST. This indicates that while the model relies more on the context, the context is not too suitable for the question, and therefore the relevance of the answer with regard to the question decreases, with a response that is more grounded in the context. Remembering the fact that llama is the only model which sticks closely to the prompt instruction in RQ1, see Section 4.1 this makes sense: llama sticks more closely to the prompt instruction to ground its responses in the given context. Therefore the faithfulness increases, but because the context is not helpful to the question, the answer relevance decreases. While mixtral and mistral stick less to the prompt instruction and therefore there is less deviation and decrease in the answer relevance compared to the setup without context and the setup with answer options, there is also less increase in the faithfulness. For all three models the setup with short NIST context seems to be especially challenging regarding the answer relevance, but for llama and mistral an increased faithfulness can be observed. This could be due to the deviation in prompt length.

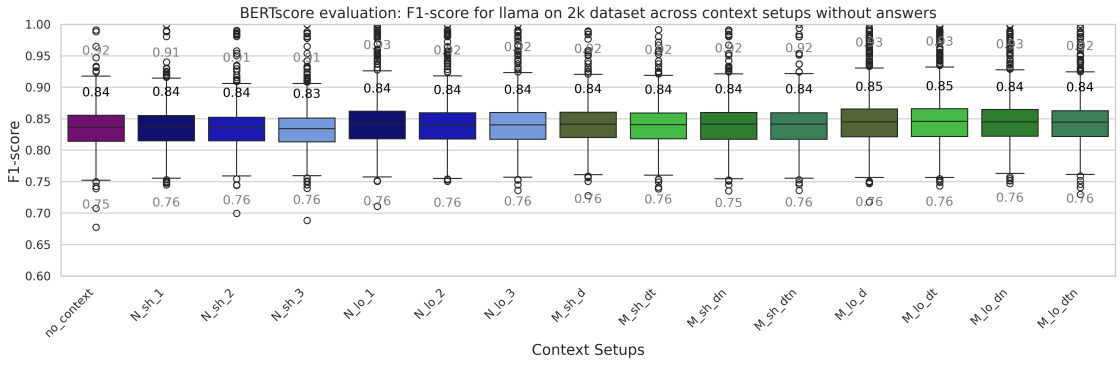
6.2.2 BERTscore

As for RQ2, see Section 5.2.3, the results of the setup without answers are evaluated with BERTscore. In this case the responses of all three models are evaluated with BERTscore. BERTscore measures the semantic similarity between two strings of text, in this case the response and the reference. The results can be found in Figure 9. The figure displays the boxplots for the average f1-scores for each model across the context setups. The detailed values for f1, precision and recall can be found in the appendix.

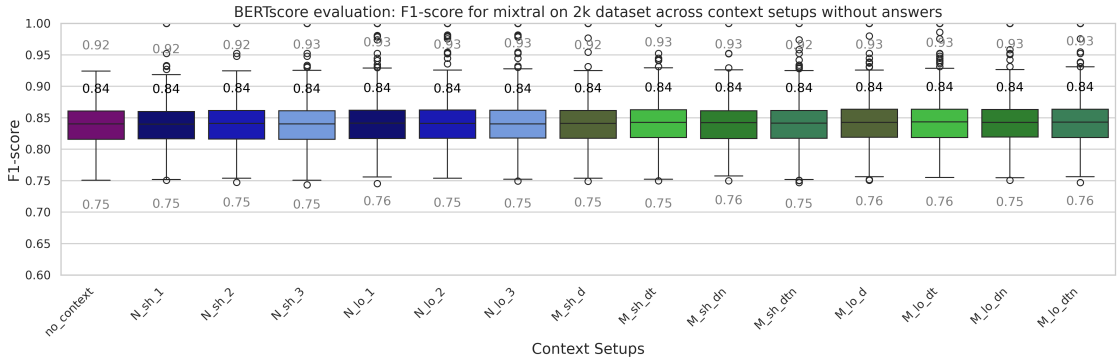
When looking at the median scores in Figure 9, across all three models and all setups it gets clear that there is no deviation. The median score is 0.84 for all of them, with three exceptions where the score is 0.83 and 0.85. The same is true for the whiskers, that all range between 0.75 and 0.76 for the lower whiskers, and 0.91 and 0.93 for the upper whiskers. Deviations are observable for the amount of outliers. The most outliers can be observed for llama, upwards and downwards. For all three models there are more outliers upwards than downwards. Mistral produces the least amount of outliers.
In general the results of BERTscore indicate a good semantic similarity between the responses and the references. For the median there is no difference observable between the performance of the setup without context and the performance of the setups with context. For mixtral there are no outliers for the setup without context, whereas for lama the setup without context produces many downward outliers. A detailed analysis of the outliers can be found in section 6.3. The exact values, including q1 and q2 and minimum and maximum values can be found in the tables in the appendix.

6.2.3 BLEUscore

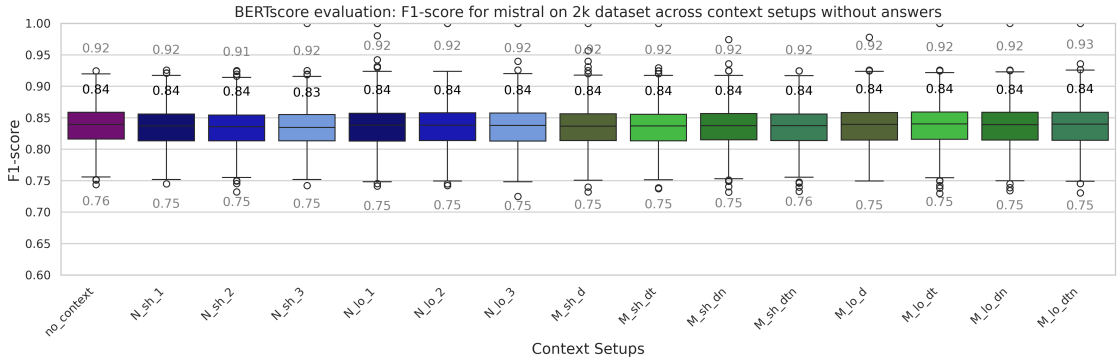
Additionally to RAGAs and BERTscore, BLEUscore is used to evaluate the results. The evaluation scores for BLEUscore can be found in Figure 10.



(a) Performance of llama

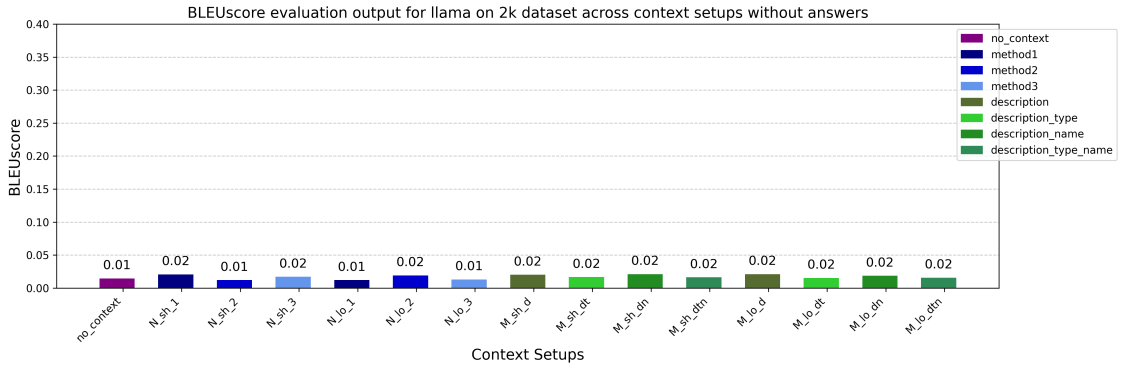


(b) Performance of mixtral

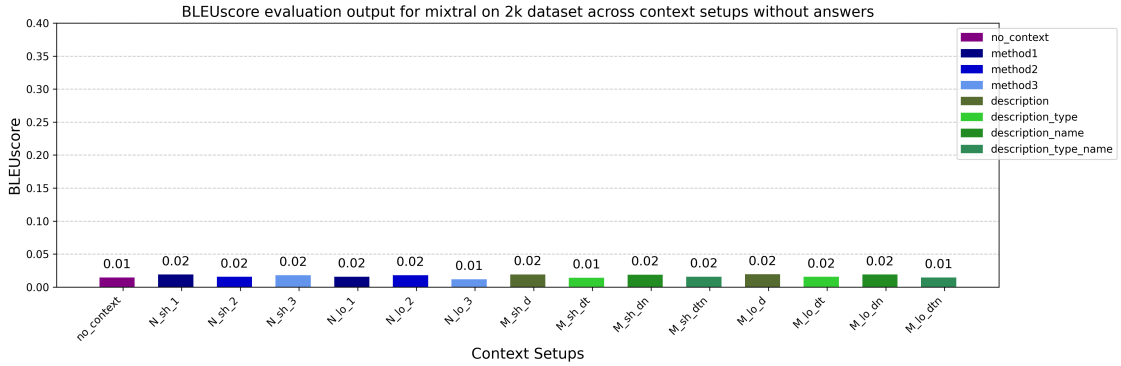


(c) Performance of mistral

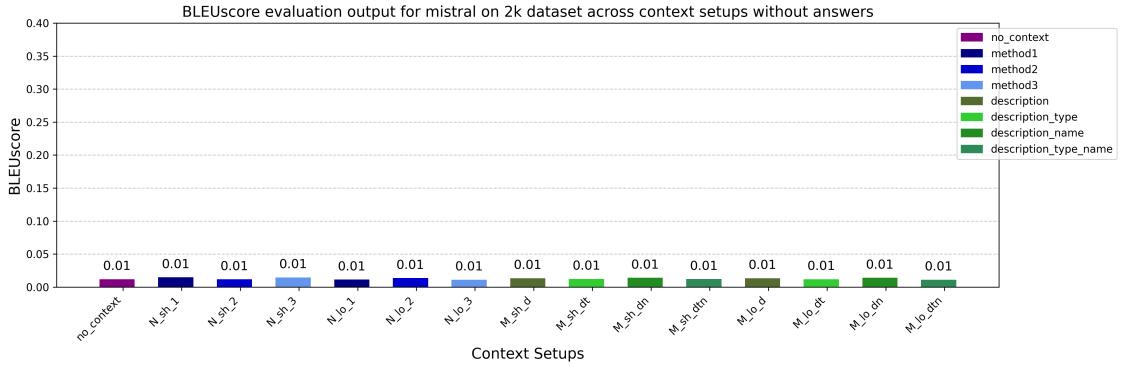
Figure 9: BERTscore evaluation across setups across models without answers: purple box shows the result for setup without context, blue boxes for the setup with NIST, green boxes for the setup with MITRE ATT&CK



(a) Performance of llama



(b) Performance of mixtral



(c) Performance of mistral

Figure 10: BLEUscore evaluation across setups across models without answers: purple bar shows the result for setup without context, blue bars for the setups with NIST, green bars for the setups with MITRE ATT&CK

When looking at the scores it can be seen that they range between 0.0.1 and 0.02. For mistral the score for every setup is 0.01. When looking at llama it can be seen that most setups yield a score of 0.02. Four setups reach a score of 0.01: the setup without context, short context based on NIST, retrieved with method2, and long context based on NIST, retrieved with method1 and method3. For mixtral the findings are similar. Most of the setups yield a score of 0.02. For four setups a score of 0.01 is yieled: the setup without context, long NIST setup retrieved with method3, short context based on MITRE ATT&CK that contains descriptions and type, and long context based on MITRE ATT&CK that contains descriptions, type and name. In general, from these numbers no impact of context length, source of context or type of context can be derived. The scores indicate a poor n-gram overlap between responses and references. Especially for mistral the BLEUscore values are worse than for RQ2 (Section 5.2.4 with the setup with answers. This makes sense because the references are taken from the correct answer options. When the answer options are not available and the model generates its response freely, it is less likely that the model's response contains exactly the terms of the reference, that was taken from the answer options. As already mentioned in RQ2 a poor BLEUscore is not an indicator for low quality responses.

6.3 Error Analysis

Outliers This section looks at the errors made for QA without answers. Since RAGAs computes a score for answer relevance and faithfulness for the whole dataset, it is not possible to look at the score for specific questions. The scores for BERTscore should be taken with caution: the model was not fine-tuned and the scores are all quite high and close to one another. Nevertheless, BERTscore can be used to investigate which questions are harder to respond to than others, and if there is a correlation between distinctive terms in the findings of RQ1, Section 4.2, and RQ2, Section 5.2, and the outliers in the current setup. Again, as for RQ2 there is a certain amount of questions that the models fail to answer at all. These issues will be

examined in the following.

The first observation that stands out when looking at the results of BERTscore in Figure 9 is that there are some scores that are higher than 1. This in theory not possible because BERTscore relies on cosine similarity which can not exceed 1.0. In the current cases the scores exceed 1.0 only minimally, for example 1.0000001192092896, and are due to computational imprecisions (Markstein, 2008).

Looking at the outliers it seems like the following questions are especially harder to respond to. They occur frequently as negative outliers:

Which algorithm was ultimately selected as the AES candidate?

To negotiate encryption keys securely over an unencrypted channel, which two methods are designed to provide this capability?

In the context of computer architecture, what is primarily responsible for integrating legacy peripheral devices and doesn2̆019t support Plug and Play (PnP) setup?

Which of the following best describes hashing?

The following questions are interesting because they occur as outliers often, but not only negative. *Which cryptographic algorithm has been standardized as the Advanced Encryption Standard (AES)?* is a negative outlier 19 times. For N_lo_2 with llama it occurs as positive outlier. *What is the term used to describe the practice of disguising a message to make it appear as normal data traffic?* occurs as outlier 14 times, whereas it is below the median eighth times, and higher than the median six times. There doesn't seem to be any pattern for the occurrences as low and high outlier. There are no questions that occur as outliers for all setups without context, and not at all for setups with context. The distinctive term *AES* falls into topic25 which occurs more often in incorrectly answered questions in RQ1, Section 4. Except from the questions mentioned above, that occur more frequently, the type of questions and their topic clusters are quite scattered across the different context setups. Therefore it can not be said clearly, that the context helps with certain questions or distinctive terms, when comparing the results of prompting without answer options.

The positive outliers are mostly cases where the response overlaps entirely with the reference, for example:

reference: "Quality of Service"

response: "Quality of Service"
f1: 0.9910897016525269
Since the setups with context do not outperform the setup without context, it makes sense that there are not less outliers and null responses in the setups with context.

setup	llama	mistral	mixtral	summary
no context	0	0	0	0
N_sh_1	0	0	0	0
N_sh_2	0	0	0	0
N_sh_3	0	0	0	0
N_lo_1	0	3	3	6
N_lo_2	0	2	2	4
N_lo_3	0	1	1	2
M_sh_d	19	19	19	57
M_sh_dt	9	9	9	27
M_sh_dn	9	9	9	27
M_sh_dtn	7	7	7	21
M_lo_d	168	168	168	504
M_lo_dt	55	55	55	165
M_lo_dn	55	55	55	165
M_lo_dtn	36	36	36	108

Table 4: Number of null responses across models and setups for setup without answers

Null Responses Table 4 shows the amount of null responses across all models for all setups for prompting without answers. There is a high overlap with the null responses produced by the models for the setup with answer options. For example *What is the primary purpose of a disaster recovery plan?* occurs in the same 24 setups. Since the null responses are most likely due to the context this makes sense, because the context stays the same, independent of the availability of answer

options. For the setup without answer options and without context there are no null responses. These observations confirms the hypothesis that some contexts are confusing to the model, leading to excess of the time restriction. Still, it should be noted that there are far more cases of null responses for the setup with answers (5305) than for the setup with free generated answers (1086) across all models and setups. This could be due to prompt length (longer with options) or more complex prompt with options that exceeds time restrictions.

6.4 Discussion

The goal of this experimental step was to examine the impact of context for a QA scenario without answer options. The evaluation with answer relevance shows that the setup without context outperforms the setups with context when no answers are available. Faithfulness is not useful to examine the difference between the setup with and without context, because it requires context for the computation of the score, and therefore no score can be computed for the setup without context. Evaluation with BLEUscore shows no difference between the setups with and without context. The same is true for BERTscore. The analysis of the outliers of the evaluation with BERTscore doesn't reveal any differences between the setup without context and the setups with context. Nevertheless, when looking at the difference between the setup with options, see Section 5, and the setup without options the following can be observed. The scores for median neither underperform nor outperform the scores of the medians for the setup with answers, meaning that according to the median the semantic similarity between response and reference is not influenced by the availability of answer options. Nevertheless, the higher frequency of upwards outliers indicates that there a more cases where there is a higher semantic similarity between the responses and the references. This is interesting because the references are taken from the answer options, so it could be assumed that the similarity would be higher if the answer options are available. When looking at the null responses it is observable that for the setup without context no null responses occur. The same is true for setups with short NIST context. In summary it can be said, that the context

doesn't increase the performance according to RAGAs, but neither decreases the performance according to BERTscore. While for the setup with answers, the answer relevance for the setups with context outperformed the setup without context, for mixtral and mistral, and for the setup without answers it didn't, the faithfulness is much higher when no answer options are available. While this observation could be due to a deviation in prompt length, it can be said that in the case, where the answer options are not available the model relies more on the context. Since the context is not ideal, the performance in answer relevance decreases, compared to the setup without context.
Since the context seems to be not very suitable for the questions, the fourth Research Question takes a deeper look into the different context setups.

7 RQ4: How do the context features impact the results?

The findings of RQ1, RQ2 and RQ3 showed that the context does not have a positive impact on the performance of the models on QA. The performance either stays approximately the same or decreases, both for the setup with and without answer options. The more the model relies on the context, the lower the performance on how well the response fits the question. Since according to literature RAG does help with model performance on QA (Es et al., 2024; Belyi et al., 2024), the current section takes a look at the context and elaborates possible reasons for the negative impact of the context.

7.1 Methdology: Evaluation of context

7.1.1 Evaluation metrics

The context is evaluated with BERTscore, BLEUscore and RAGAs' context relevance. BERTscore computes the semantic similarity between two strings of text,

in this case between the paragraphs in the context and the question. BLEUscore computes the n-gram overlap between two strings of text, in this case between the paragraphs in the context and the question. RAGAs' context relevance computes how relevant the context is compared to the question.

7.1.2 Evaluation setup

As already mentioned the context will be evaluated with regard to the questions. This means, that BERTscore and BLEUscore will be used as comparison towards context relevance, not as comparison towards faithfulness, which evaluates the response with regard to the context. Since the context stays the same for every model there are 14 setups to be evaluated.

7.2 Results

The results for all three metrics evaluating the context can be seen in Figure 11. First the results for context relevance will be discussed, followed by the evaluation with BERTscore and finally the evaluation with BLEUscore.

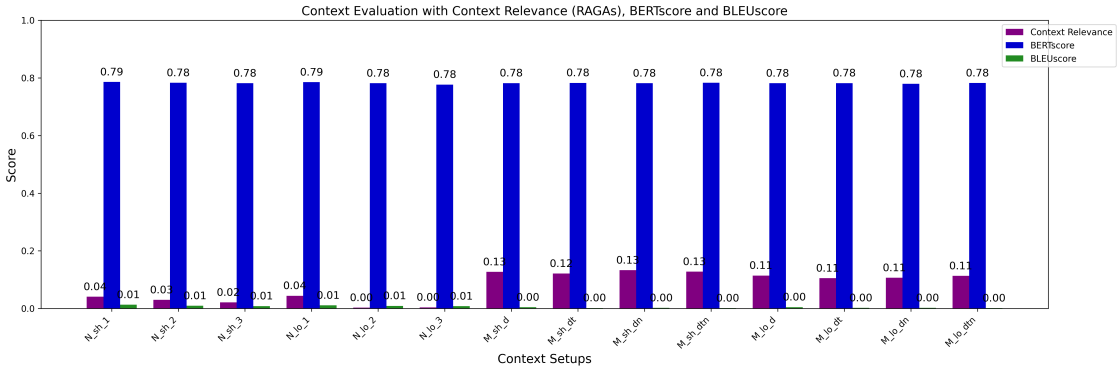


Figure 11: Context Evaluation with Context Relevance, BERTscore and BLEUscore. Purple bars show the context relevance, blue bars the BERTscore and green bars the BLEUscore. The scores for BERT and BLEU are their median f1-scores across the whole dataset

7.2.1 Context Relevance

As can be seen in Figure 11 there is a clear observation for the evaluation with context relevance: the context retrieved from MITRE ATT&CK yields an average score of 0.119, while the context retrieved from NIST yields an average score of 0.022. It can clearly be said, that the context retrieved from MITRE ATT&CK is more relevant to the question than the context retrieved from NIST. This is most probably due to the source of context.
When looking at the context length, the short context yields better results for both NIST and MITRE ATT&CK . The short NIST context reaches an average score of 0.03, compared to an average score of 0.013. It is important to note that the long NIST contexts retrieved with method2 and method3 yield a context relevance score of 0. For MITRE ATT&CK the long context reaches an average score of 0.128, compared to an average score of 0.11 for the long context. To sum it up, it can be said, that longer context is not necessarily better and data retrieved from MITRE ATT&CK serves better as context than data retrieved from NIST. This confirms the observation from RQ1 (Section 4), RQ2 (Section 5) and RQ3 (Section 6) that in most cases the setups based on MITRE ATT&CK outperform the ones based on NIST. Nevertheless, all the scores are quite low and indicate that the context is not relevant to the questions, which is also coherent with the observations from before.

7.2.2 BERTscore

The results of the evaluation with BERTscore show a score of 0.78 for all setups except for the short and long NIST context retrieved with method1, which yield a score of 0.79. This continuity within the scores for BERTscore was already observed for RQ2 and RQ3 and may be due to not fine-tuning the model. The scores here suggest a good semantic similarity between the context paragraphs and the questions, although they are not ideal. Method1 for the context based on NIST has slightly higher scores, which seems reasonable, because method1 retrieves the x paragraphs that are closest to the question, measured with cosine simlarity.

7.2.3 BLEUscore

When looking at the evaluation with BLEUscore the following can be observed. For all setups based on MITRE ATT&CK the BLEUscore is 0. For all setups based on NIST the BLEUscore reaches a score of 0.01. Altogether these scores indicate a very low n-gram overlap between the context paragraphs and the questions. One reason why the scores for context based on NIST are higher could lie in the data structure: the NISTs database is built from annual reports which mainly consist of sentences, while the MITRE ATT&CK database is built on a knowledge base in the form of a tree structure, that contains more single words, as for example the type and the name. Therefore the n-gram overlap might be higher for the data built on whole sentences.

While on average there is a higher n-gram overlap and semantic similarity for the context based on NIST, context relevance shows higher scores for the context based on MITRE ATT&CK and RQ2 and RQ3 show better performance for the setups based on MITRE ATT&CK . This suggests that a close semantic similarity or a high overlap of exact terms is no guarantee for good context.

7.3 Distinctive Terms

As was already discussed in RQ1, RQ2 and RQ3, see Section 4.2, 5.2 and 6.2 respectively, the context does not necessarily help with answering the questions. The goal of this subsection is to investigate whether certain context setups have an especially negative impact on the model performance. The BERTopic analysis in Section 4 showed that some topics are especially hard to answer for the models. Therefore this section takes a look at the topic clusters again.
For the setup with answer options there are seven questions where in 24 out of 42 setups the models did not respond. These questions are:
1. *What is the primary purpose of a disaster recovery plan?*
2. *What is the primary goal of the disaster recovery plan during a business disruption?*

3. *In continuity planning, what is the significance of a recovery point objective (RPO)?*
4. *What is the primary objective of a disaster recovery plan in relation to physical security?*
5. *What is the primary goal of disaster recovery planning?*
6. *According to the principles of continuity planning, what should be facilitated during recovery strategy development?*
7. *What is the main purpose of a disaster recovery plan (DRP)?*
Further, there are two questions that weren't responded to by the models in 21 out of 42 cases:
8. *In the context of HIPAA information security requirements, which HIPAA-CMM practice focuses on developing disaster recovery and business continuity plans?*
9. *Which version of IIS is commonly encountered in the wild for Windows Server 2008?*
And in 17 cases:
10. *What security method, mechanism, or model reveals a capabilities list of a subject across multiple objects?*
Question 5 appears in the incorrectly answered questions for all models for the setup without context. The others don't.
Interestingly, for some questions in certain setups the exact same context is retrieved. For example for
What is the primary goal of disaster recovery planning?, *What is the primary purpose of a disaster recovery plan?* and *What is the main purpose of a disaster recovery plan (DRP)?* for short context based on MITRE ATT&CK that contains descriptions and type. For *What does the Object-Oriented Security Model (OOSM) focus on?*, *What does the *-property in the Biba model indicate?*, *What security method, mechanism, or model reveals a capabilities list of a subject across multiple objects?* the exact same context is retrieved for short context based on MITRE ATT&CK that contains only descriptions. This can be due to the retrieval method: the mentioned questions are similar, therefore they are located in the same area in the vector space and it is likely that the same x closest context entities are retrieved.

Apart from question 6, 9 and 10, the questions above all belong to topic cluster 43. The questions from this cluster appear to be hard to answer for the model with and without context.
The analysis of the context reveals that it is not due to the length of the context that the model fails to respond to the questions. As can be seen in the evaluation with context relevance, some context setups are certainly better than others and there is no clear answer to why for some contexts the models fail to answer. Most likely, answering the questions exceeds the time restrictions of responding to a prompt. Since the performance decreases with context, it could be said that the context distracts and confuses the model, and therefore the model takes longer to answer the questions, which leads to more null responses.
When looking at the null responses produced by the setup without answer options, findings show that they are overlapping with the null responses produced by the setup with answer options. While there are far less null responses, all null responses that occur for the setup without answer options, also occur as null responses for the setup with answer options.
Interestingly, while the evaluation of the model responses and the separate evaluation of the context show that information retrieved from MITRE ATT&CK serves as the better context, there occur far more null responses in the context setups based on MITRE ATT&CK . Although, for the setup with answer options there are not more incorrect answers for the context based on MITRE ATT&CK , the percentage of null responses is much higher. For the setup without answer options it is hard to count the "incorrect" questions, but there too, context based on MITRE ATT&CK delivers much more null responses. The context is not necessarily longer, but maybe more complex leading to an exceeding of the time restriction when the model tries to respond to the question while taking the context into account. Notably, although there are more null responses, the context based on MITRE ATT&CK still performs better.

Since the performance for the setup with context decreases, there are more incorrectly answered questions. Therefore it can not be said, that the context helps with

certain topic clusters. The amount of incorrectly answered questions increases for all clusters.
For the setup without answer options, where there are not incorrect answers, but outliers, there are two questions out of topic25 that occur with increased frequency as negative outliers. Those questions are:
1. Which algorithm was ultimately selected as the AES candidate?
2. Which cryptographic algorithm has been standardized as the Advanced Encryption Standard (AES)?
These two questions seem to be hard to respond to by the models, when no answer options are available, independent of the availability of additional context.

7.4 Discussion

In summary it can be said that the context based on a database that includes cyberattacks, the corresponding detection and mitigation strategies serves better augmentation than the context based on annual reports. Nevertheless, at least for the setup with answer options, the setup without any context yields the best results. For the setup without answer options, the setups with context partially outperform the setup without context. It can be said, that while the additional context is misleading and distracting to the model when there are answer options available, it can help when no answer options are available. It should be noted though, that with regard to answer relevance the setup without answer options performs better than the setup with answer options. This could be due to prompt length or not ideal answer options. Additionally it was discovered that in a lot of cases the performance was impacted by how closely the model followed the prompt instruction.

8 Conclusion and Future Work

The current work examined the impact of RAG on the performance of LLMs in a QA scenario in the cybersecurity domain. Therefore for each question context was retrieved from different databases in different manners and prompted to the model

with the question - first with answer options, then without answer options. It was shown that the context helps only to a certain degree, and that the relevance of the answer decreases, when the model relies more on the context. This indicates that the context is not helpful for answering the questions. This can be due to different things and leads to open questions for future work. The first issue for the unsuitable context can lie in the retrieval method. tf-idf retrieval is a simple and basic retrieval method. It is good to start with a less complex and highly interpretable retrieval method, but more advanced techniques may improve the resulting context, for example the one that Es et al. (2024) use. Another possible reason lies in the type of database. NIST was shown to not work well, while MITRE ATT&CK performed well, but not ideal. Searching for other data sources for building the databases, or changing the type of information that is retrieved from them, could shed some light. Lastly, one parameter that may have caused performance differences is the deviating prompt length, due to excluding context and/or answer options. Finding solutions for better alignment of the context length between the different setups, could lead to new findings. Lastly, the compliance of the models with the prompt instruction seems to have had a high impact on the performance. Since RAG was shown to improve the performance in QA in the past, changing these parameters could lead to an improvement of performance and therefore make QA systems more reliable and trustworthy in the cybersecurity domain.

9 Appendix

Acronyms

AI Artifical Intelligence. 10, 11

API Application Programming Interface. 19, 47

AWS Amazon Web Services. 19

BERT Bidirectional Encoder Representations from Transformers. 11

idf inverted document frequency. 13

IR Information Retrieval. 6

LLM Large Language Model. 1–3, 5–12, 14, 17, 18, 27, 31, 45, 46, 66

NER Named Entity Recognition. 6, 10

NIST National Institute for Standard Technology. 28, 29, 31, 33–36, 38–42, 45, 49, 51, 52, 54–56, 59, 62, 63, 67

NLP Natural Language Processing. 6, 9

POS-Tagging Part-of-Speech-Tagging. 6

QA Question Answering. 3, 6, 7, 9, 11, 12, 17, 18, 27, 45, 47, 49, 56, 59, 60, 66, 67

RAG Retrieved Augmented Generation. 3, 7, 11, 12, 17, 27–29, 34, 43, 60, 66, 67

RAGAs Retrieval Augmented Generation Assessment. 3, 15, 16, 32, 36, 46, 48, 49, 53, 56, 60, 61

STIX Structured Threat Information Expression. 28

tf term frequency. 13

tf-idf Term frequency inverse document frequency. 12–14, 20, 29, 31, 67

List of Figures

List of Tables

Tables

Table 5: Frequency of topic clusters: 2k dataset compared to incorrectly answered questions for all three models for setup with answer options and without context

	all data	**llama**	**mixtral**	**mistral**	**name**
-1	0.339	0.356	0.393	0.356	-1_the_what_is_of
0	0.064	0.093	0.088	0.089	0_security_organization_and_standard
1	0.044	0.037	0.032	0.042	1_authentication_password_biometric_passwords
2	0.042	0.04	0.036	0.04	2_access_control_least_privilege
3	0.029	0.028	0.032	0.028	3_risk_assessment_you_management
4	0.029	0.031	0.023	0.031	4_malware_virus_malicious_type
5	0.025	0.015	0.019	0.017	5_privacy_unauthorized_information_term
6	0.022	0.034	0.032	0.029	6_intrusion_detection_ids_activity
7	0.021	0.012	0.016	0.017	7_firewall_firewalls_rule_network
8	0.021	0.034	0.029	0.027	8_forensics_evidence_forensic_computer
9	0.021	0.015	0.023	0.027	9_scan_sniffing_tool_network
10	0.018	0.012	0.013	0.025	10_wireless_network_communication_technology
11	0.018	0.019	0.016	0.027	11_key_encryption_decryption_both
12	0.018	0.022	0.016	0.022	12_protocol_transport_layer_packet
13	0.017	0.009	0.003	0.006	13_encryption_cryptography_main_purpose
14	0.015	0.031	0.023	0.023	14_cloud_computing_environment_without
15	0.015	0.012	0.023	0.01	15_public_key_certificate_pki
16	0.014	0.009	0.023	0.012	16_social_engineering_attacks_target
17	0.013	0.022	0.006	0.012	17_phishing_click_cybersecurity_fraud
18	0.013	0.003	0.00	0.006	18_audit_monitoring_cybersecurity_purpose
19	0.012	0.012	0.016	0.008	19_attack_machine_dos_attacker
20	0.011	0.00	0.013	0.008	20_dns_domain_record_spoofing
21	0.011	0.003	0.013	0.01	21_hash_digital_hashing_message
22	0.01	0.006	0.003	0.006	22_awareness_training_program_security
23	0.01	0.006	0.01	0.006	23_wireless_access_point_points
24	0.009	0.009	0.006	0.009	24_ip_addresses_address_ipv6
25	0.009	0.022	0.00	0.012	25_transposition_aes_columnar_linear
26	0.009	0.015	0.006	0.01	26_unauthorized_measure_prevent_network

Continued on next page

	all data	llama	mixtral	mistral	name
27	0.009	0.006	0.01	0.006	27_continuity_business_planning_plan
28	0.009	0.009	0.01	0.009	28_osi_layer_model_primarily
29	0.008	0.00	0.003	0.006	29_vpn_virtual_private_vpns
30	0.007	0.00	0.003	0.008	30_software_source_updates_closed
31	0.007	0.006	0.003	0.002	31_database_relational_sql_transactions
32	0.007	0.006	0.01	0.006	32_backup_backups_data_process
33	0.007	0.006	0.003	0.006	33_ethical_hacker_hacking_hackers
34	0.007	0.006	0.006	0.006	34_honeypot_spam_botnet_bots
35	0.007	0.006	0.003	0.006	35_injection_sql_attack_ldap
36	0.006	0.003	0.01	0.002	36_hipaa_medical_privacy_healthcare
37	0.006	0.009	0.013	0.009	37_ipsec_ike_stand_associations
38	0.006	0.00	0.003	0.002	38_steganography_file_watermarking_overt
39	0.006	0.006	0.003	0.006	39_fault_tolerance_safetycritical_raid
40	0.006	0.006	0.003	0.01	40_configuration_management_change_ccb
41	0.005	0.003	0.00	0.00	41_penetration_test_testing_reconnaissance
42	0.005	0	0	0	42_incident_response_objective_organizations
43	0.005	0.003	0.003	0.003	43_disaster_recovery_plan_developing
44	0.005	0.012	0.006	0.01	44_memory_overflow_timememory_buffer

Table 6: BERTscore evaluation with boxplots for mixtral on 2k dataset with answers

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
no_context	f1	2000	0.736	0.736	0.805	0.835	0.864	0.945	0.956
N_sh_1	f1	2000	0.755	0.755	0.814	0.838	0.866	0.923	0.923
N_sh_2	f1	2000	0.745	0.745	0.814	0.839	0.866	0.926	0.926
N_sh_3	f1	2000	0.735	0.754	0.814	0.838	0.866	0.924	0.924
N_lo_1	f1	2000	0.746	0.746	0.816	0.842	0.87	0.931	0.931
N_lo_2	f1	2000	0.757	0.757	0.815	0.841	0.87	0.939	0.939

Continued on next page

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
N_lo_3	f1	2000	0.755	0.755	0.816	0.84	0.87	0.939	0.939
M_sh_d	f1	2000	0.736	0.736	0.812	0.835	0.865	0.925	0.925
M_sh_dt	f1	2000	0.736	0.736	0.812	0.836	0.863	0.921	0.921
M_sh_dnn	f1	2000	0.736	0.736	0.812	0.836	0.863	0.918	0.918
M_sh_dtn	f1	2000	0.741	0.741	0.812	0.836	0.862	0.919	0.919
M_lo_d	f1	2000	0.737	0.754	0.815	0.837	0.864	0.925	0.939
M_lo_dt	f1	2000	0.757	0.757	0.815	0.838	0.867	0.923	0.923
M_lo_dn	f1	2000	0.736	0.757	0.814	0.839	0.865	0.925	0.925
M_lo_dtn	f1	2000	0.736	0.762	0.814	0.838	0.866	0.931	0.931
no_context	precision	2000	0.692	0.713	0.779	0.799	0.825	0.895	0.935
N_sh_1	precision	2000	0.728	0.728	0.782	0.803	0.826	0.888	0.888
N_sh_2	precision	2000	0.743	0.743	0.781	0.802	0.826	0.889	0.896
N_sh_3	precision	2000	0.725	0.725	0.781	0.802	0.825	0.884	0.895
N_lo_1	precision	2000	0.715	0.735	0.784	0.806	0.83	0.898	0.898
N_lo_2	precision	2000	0.734	0.734	0.784	0.806	0.83	0.899	0.917
N_lo_3	precision	2000	0.735	0.735	0.783	0.805	0.829	0.891	0.917
M_sh_d	precision	2000	0.727	0.727	0.777	0.799	0.823	0.89	0.89
M_sh_dt	precision	2000	0.732	0.732	0.779	0.798	0.821	0.882	0.884
M_sh_dnn	precision	2000	0.731	0.731	0.778	0.797	0.821	0.884	0.887
M_sh_dtn	precision	2000	0.716	0.716	0.778	0.798	0.82	0.883	0.883
M_lo_d	precision	2000	0.722	0.722	0.781	0.806	0.827	0.888	0.917
M_lo_dt	precision	2000	0.705	0.737	0.781	0.803	0.826	0.887	0.896
M_lo_dn	precision	2000	0.729	0.729	0.78	0.802	0.825	0.888	0.895
M_lo_dtn	precision	2000	0.742	0.742	0.781	0.802	0.825	0.888	0.901
no_context	recall	2000	0.699	0.728	0.836	0.874	0.91	0.98	0.98
N_sh_1	recall	2000	0.703	0.747	0.848	0.879	0.916	0.968	0.968
N_sh_2	recall	2000	0.699	0.754	0.85	0.88	0.915	0.969	0.969
N_sh_3	recall	2000	0.697	0.758	0.849	0.879	0.914	0.963	0.963
N_lo_1	recall	2000	0.733	0.752	0.85	0.882	0.918	0.967	0.967

Continued on next page

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
N_lo_2	recall	2000	0.733	0.746	0.849	0.882	0.918	0.969	0.969
N_lo_3	recall	2000	0.737	0.75	0.85	0.882	0.919	0.972	0.972
M_sh_d	recall	2000	0.699	0.747	0.848	0.879	0.915	0.967	0.967
M_sh_dt	recall	2000	0.699	0.753	0.849	0.879	0.913	0.964	0.964
M_sh_dnn	recall	2000	0.699	0.754	0.849	0.88	0.914	0.961	0.961
M_sh_dtn	recall	2000	0.741	0.757	0.848	0.88	0.913	0.963	0.963
M_lo_d	recall	2000	0.718	0.735	0.841	0.877	0.913	0.967	0.967
M_lo_dt	recall	2000	0.744	0.747	0.848	0.88	0.916	0.964	0.964
M_lo_dn	recall	2000	0.699	0.75	0.847	0.881	0.914	0.966	0.966
M_lo_dtn	recall	2000	0.697	0.754	0.849	0.881	0.914	0.969	0.969

Table 7: BERTscore evaluation with boxplots for mistral on 2k dataset with answers

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
no_context	f1	2000	0.766	0.766	0.827	0.862	0.896	0.964	0.964
N_sh_1	f1	2000	0.75	0.75	0.82	0.846	0.879	0.964	0.964
N_sh_2	f1	2000	0.742	0.742	0.819	0.844	0.877	0.961	0.967
N_sh_3	f1	2000	0.738	0.738	0.819	0.843	0.874	0.956	0.961
N_lo_1	f1	2000	0.754	0.754	0.821	0.853	0.895	0.964	0.964
N_lo_2	f1	2000	0.763	0.763	0.823	0.853	0.896	0.972	0.972
N_lo_3	f1	2000	0.753	0.753	0.824	0.852	0.894	0.97	0.97
M_sh_d	f1	2000	0.75	0.75	0.82	0.845	0.876	0.96	0.961
M_sh_dt	f1	2000	0.758	0.758	0.818	0.844	0.873	0.957	0.961

Continued on next page

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
M_sh_dnn	f1	2000	0.746	0.746	0.818	0.843	0.872	0.952	0.961
M_sh_dtn	f1	2000	0.758	0.758	0.816	0.841	0.872	0.956	0.961
M_lo_d	f1	2000	0.755	0.755	0.823	0.848	0.884	0.964	0.964
M_lo_dt	f1	2000	0.75	0.75	0.823	0.849	0.884	0.961	0.961
M_lo_dn	f1	2000	0.761	0.761	0.822	0.85	0.882	0.961	0.961
M_lo_dtn	f1	2000	0.736	0.736	0.821	0.85	0.882	0.963	0.963
no_context	precision	2000	0.738	0.738	0.802	0.833	0.866	0.953	0.953
N_sh_1	precision	2000	0.727	0.727	0.793	0.816	0.844	0.921	0.953
N_sh_2	precision	2000	0.727	0.727	0.792	0.814	0.841	0.915	0.96
N_sh_3	precision	2000	0.735	0.735	0.791	0.813	0.839	0.911	0.946
N_lo_1	precision	2000	0.723	0.723	0.796	0.826	0.863	0.952	0.952
N_lo_2	precision	2000	0.742	0.742	0.797	0.826	0.865	0.959	0.959
N_lo_3	precision	2000	0.735	0.735	0.797	0.825	0.862	0.958	0.958
M_sh_d	precision	2000	0.726	0.726	0.79	0.814	0.84	0.914	0.946
M_sh_dt	precision	2000	0.734	0.734	0.789	0.811	0.835	0.904	0.943
M_sh_dnn	precision	2000	0.734	0.734	0.789	0.81	0.835	0.904	0.944
M_sh_dtn	precision	2000	0.728	0.728	0.787	0.81	0.833	0.903	0.943
M_lo_d	precision	2000	0.707	0.733	0.798	0.824	0.852	0.93	0.949
M_lo_dt	precision	2000	0.742	0.742	0.796	0.821	0.852	0.934	0.946
M_lo_dn	precision	2000	0.738	0.738	0.795	0.82	0.851	0.934	0.946
M_lo_dtn	precision	2000	0.74	0.74	0.794	0.819	0.849	0.93	0.948
no_context	recall	2000	0.754	0.754	0.852	0.893	0.932	0.982	0.982
N_sh_1	recall	2000	0.703	0.737	0.847	0.88	0.922	0.983	0.983
N_sh_2	recall	2000	0.732	0.737	0.847	0.88	0.922	0.983	0.983
N_sh_3	recall	2000	0.696	0.749	0.846	0.88	0.919	0.982	0.982
N_lo_1	recall	2000	0.738	0.738	0.847	0.885	0.932	0.983	0.983
N_lo_2	recall	2000	0.741	0.741	0.849	0.886	0.932	0.985	0.985
N_lo_3	recall	2000	0.732	0.732	0.849	0.885	0.931	0.983	0.983
M_sh_d	recall	2000	0.727	0.741	0.849	0.882	0.921	0.983	0.983

Continued on next page

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
M_sh_dt	recall	2000	0.746	0.746	0.848	0.884	0.92	0.981	0.981
M_sh_dnn	recall	2000	0.723	0.739	0.846	0.882	0.918	0.982	0.982
M_sh_dtn	recall	2000	0.751	0.751	0.844	0.88	0.919	0.981	0.981
M_lo_d	recall	2000	0.736	0.736	0.842	0.882	0.924	0.983	0.983
M_lo_dt	recall	2000	0.731	0.732	0.848	0.886	0.925	0.982	0.982
M_lo_dn	recall	2000	0.737	0.737	0.847	0.886	0.924	0.984	0.984
M_lo_dtn	recall	2000	0.719	0.749	0.849	0.885	0.925	0.983	0.983

Table 8: BERTscore evaluation with boxplots for llama on 2k dataset without answers

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
no_context	f1	1997	0.678	0.752	0.814	0.836	0.856	0.918	0.991
N_sh_1	f1	1997	0.745	0.756	0.815	0.837	0.855	0.914	1.0
N_sh_2	f1	1997	0.7	0.759	0.815	0.836	0.852	0.906	0.991
N_sh_3	f1	1997	0.688	0.76	0.813	0.834	0.851	0.906	1.0
N_lo_1	f1	1997	0.711	0.758	0.818	0.842	0.862	0.926	1.0
N_lo_2	f1	1997	0.751	0.755	0.818	0.84	0.86	0.918	1.0
N_lo_3	f1	1997	0.736	0.757	0.817	0.84	0.86	0.923	1.0
M_sh_d	f1	1978	0.728	0.761	0.82	0.841	0.86	0.92	0.989
M_sh_dt	f1	1988	0.739	0.76	0.818	0.84	0.859	0.919	0.991
M_sh_dn	f1	1988	0.736	0.755	0.817	0.841	0.86	0.921	0.991
M_sh_dtn	f1	1990	0.736	0.756	0.817	0.841	0.859	0.922	1.0
M_lo_d	f1	1829	0.718	0.757	0.822	0.845	0.865	0.93	1.0

Continued on next page

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
M_lo_dt	f1	1942	0.743	0.757	0.822	0.846	0.866	0.932	1.0
M_lo_dn	f1	1942	0.747	0.763	0.822	0.845	0.865	0.928	1.0
M_lo_dtn	f1	1961	0.73	0.762	0.822	0.845	0.863	0.925	1.0
no_context	precision	1997	0.615	0.73	0.792	0.812	0.834	0.897	0.992
N_sh_1	precision	1997	0.718	0.75	0.802	0.824	0.842	0.898	1.0
N_sh_2	precision	1997	0.622	0.744	0.802	0.823	0.84	0.896	0.992
N_sh_3	precision	1997	0.627	0.752	0.802	0.821	0.838	0.891	1.0
N_lo_1	precision	1997	0.627	0.743	0.806	0.829	0.849	0.912	1.0
N_lo_2	precision	1997	0.72	0.752	0.804	0.827	0.845	0.907	1.0
N_lo_3	precision	1997	0.725	0.756	0.805	0.828	0.846	0.906	1.0
M_sh_d	precision	1978	0.744	0.75	0.804	0.826	0.844	0.903	0.991
M_sh_dt	precision	1988	0.75	0.75	0.803	0.825	0.844	0.905	0.992
M_sh_dn	precision	1988	0.749	0.749	0.804	0.826	0.844	0.905	0.992
M_sh_dtn	precision	1990	0.739	0.752	0.803	0.825	0.844	0.905	1.0
M_lo_d	precision	1829	0.744	0.744	0.808	0.831	0.851	0.913	1.0
M_lo_dt	precision	1942	0.738	0.757	0.807	0.831	0.851	0.916	1.0
M_lo_dn	precision	1942	0.748	0.748	0.808	0.831	0.85	0.911	1.0
M_lo_dtn	precision	1961	0.731	0.755	0.807	0.831	0.848	0.907	1.0
no_context	recall	1997	0.729	0.766	0.836	0.861	0.883	0.953	0.99
N_sh_1	recall	1997	0.704	0.759	0.827	0.851	0.873	0.941	1.0
N_sh_2	recall	1997	0.704	0.763	0.827	0.849	0.87	0.935	0.989
N_sh_3	recall	1997	0.706	0.76	0.824	0.848	0.868	0.933	1.0
N_lo_1	recall	1997	0.706	0.758	0.831	0.856	0.88	0.954	1.0
N_lo_2	recall	1997	0.708	0.756	0.829	0.854	0.878	0.951	1.0
N_lo_3	recall	1997	0.696	0.757	0.829	0.854	0.877	0.949	1.0
M_sh_d	recall	1978	0.691	0.76	0.832	0.857	0.88	0.95	0.988
M_sh_dt	recall	1988	0.705	0.758	0.831	0.856	0.88	0.95	0.991
M_sh_dn	recall	1988	0.701	0.759	0.831	0.856	0.88	0.952	0.99
M_sh_dtn	recall	1990	0.703	0.76	0.832	0.856	0.88	0.949	1.0

Continued on next page

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
M_lo_d	recall	1829	0.689	0.76	0.834	0.86	0.884	0.958	1.0
M_lo_dt	recall	1942	0.699	0.759	0.834	0.86	0.885	0.961	1.0
M_lo_dn	recall	1942	0.701	0.763	0.835	0.859	0.883	0.955	1.0
M_lo_dtn	recall	1961	0.701	0.76	0.833	0.859	0.882	0.954	1.0

Table 9: BERTscore evaluation with boxplots for mixtral on 2k dataset without answers

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
no_context	f1	1997	0.751	0.751	0.816	0.84	0.861	0.924	0.924
N_sh_1	f1	1997	0.751	0.752	0.816	0.84	0.86	0.919	1.0
N_sh_2	f1	1997	0.748	0.754	0.816	0.841	0.861	0.925	1.0
N_sh_3	f1	1997	0.744	0.751	0.816	0.84	0.861	0.925	1.0
N_lo_1	f1	1994	0.746	0.756	0.817	0.841	0.862	0.929	1.0
N_lo_2	f1	1995	0.754	0.754	0.817	0.841	0.862	0.926	1.0
N_lo_3	f1	1996	0.75	0.752	0.818	0.84	0.862	0.928	1.0
M_sh_d	f1	1978	0.749	0.754	0.817	0.841	0.861	0.925	1.0
M_sh_dt	f1	1988	0.75	0.752	0.819	0.842	0.863	0.929	1.0
M_sh_dn	f1	1988	0.75	0.758	0.817	0.842	0.861	0.926	1.0
M_sh_dtn	f1	1990	0.747	0.752	0.817	0.842	0.862	0.925	1.0
M_lo_d	f1	1829	0.751	0.756	0.819	0.843	0.863	0.926	1.0
M_lo_dt	f1	1942	0.755	0.755	0.819	0.843	0.863	0.928	1.0
M_lo_dn	f1	1942	0.751	0.755	0.819	0.843	0.863	0.926	1.0
M_lo_dtn	f1	1961	0.747	0.756	0.819	0.843	0.864	0.931	1.0

Continued on next page

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
no_context	precision	1997	0.739	0.739	0.796	0.819	0.841	0.904	0.923
N_sh_1	precision	1997	0.742	0.742	0.803	0.823	0.843	0.903	1.0
N_sh_2	precision	1997	0.742	0.742	0.802	0.823	0.844	0.905	1.0
N_sh_3	precision	1997	0.742	0.742	0.8	0.823	0.844	0.907	1.0
N_lo_1	precision	1994	0.727	0.737	0.801	0.825	0.845	0.911	1.0
N_lo_2	precision	1995	0.737	0.737	0.801	0.824	0.844	0.906	1.0
N_lo_3	precision	1996	0.742	0.742	0.8	0.824	0.845	0.906	1.0
M_sh_d	precision	1978	0.745	0.745	0.801	0.824	0.844	0.905	1.0
M_sh_dt	precision	1988	0.734	0.749	0.802	0.824	0.845	0.909	1.0
M_sh_dn	precision	1988	0.744	0.744	0.801	0.824	0.844	0.905	1.0
M_sh_dtn	precision	1990	0.719	0.747	0.801	0.824	0.845	0.909	1.0
M_lo_d	precision	1829	0.742	0.742	0.803	0.825	0.846	0.903	1.0
M_lo_dt	precision	1942	0.744	0.744	0.803	0.826	0.846	0.906	1.0
M_lo_dn	precision	1942	0.738	0.747	0.804	0.826	0.846	0.907	1.0
M_lo_dtn	precision	1961	0.749	0.749	0.803	0.826	0.847	0.913	1.0
no_context	recall	1997	0.721	0.761	0.835	0.863	0.886	0.961	0.97
N_sh_1	recall	1997	0.71	0.756	0.83	0.857	0.881	0.952	1.0
N_sh_2	recall	1997	0.715	0.753	0.831	0.858	0.883	0.96	1.0
N_sh_3	recall	1997	0.71	0.753	0.83	0.858	0.882	0.958	1.0
N_lo_1	recall	1994	0.71	0.753	0.832	0.859	0.884	0.962	1.0
N_lo_2	recall	1995	0.706	0.754	0.832	0.858	0.884	0.962	1.0
N_lo_3	recall	1996	0.716	0.759	0.833	0.86	0.883	0.956	1.0
M_sh_d	recall	1978	0.723	0.761	0.833	0.86	0.882	0.953	1.0
M_sh_dt	recall	1988	0.72	0.761	0.834	0.861	0.884	0.956	1.0
M_sh_dn	recall	1988	0.72	0.759	0.833	0.86	0.883	0.953	1.0
M_sh_dtn	recall	1990	0.703	0.76	0.833	0.86	0.883	0.954	1.0
M_lo_d	recall	1829	0.724	0.759	0.834	0.861	0.885	0.962	1.0
M_lo_dt	recall	1942	0.726	0.754	0.833	0.861	0.885	0.962	1.0
M_lo_dn	recall	1942	0.716	0.76	0.834	0.86	0.884	0.958	1.0

Continued on next page

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
M_lo_dtn	recall	1961	0.725	0.757	0.833	0.861	0.884	0.96	1.0

Table 10: BERTscore evaluation with boxplots for mistral on 2k dataset without answers

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
no_context	f1	1997	0.744	0.756	0.816	0.839	0.859	0.92	0.925
N_sh_1	f1	1997	0.745	0.752	0.813	0.837	0.856	0.917	0.926
N_sh_2	f1	1997	0.732	0.755	0.813	0.836	0.854	0.914	0.925
N_sh_3	f1	1997	0.742	0.752	0.813	0.834	0.855	0.916	1.0
N_lo_1	f1	1994	0.741	0.748	0.813	0.838	0.857	0.924	1.0
N_lo_2	f1	1995	0.742	0.75	0.814	0.838	0.858	0.924	1.0
N_lo_3	f1	1996	0.725	0.748	0.813	0.838	0.857	0.92	1.0
M_sh_d	f1	1978	0.732	0.751	0.814	0.837	0.856	0.918	1.0
M_sh_dt	f1	1988	0.738	0.751	0.814	0.837	0.856	0.917	1.0
M_sh_dn	f1	1988	0.732	0.753	0.815	0.837	0.857	0.917	0.974
M_sh_dtn	f1	1990	0.733	0.755	0.814	0.837	0.856	0.917	0.925
M_lo_d	f1	1829	0.749	0.749	0.815	0.84	0.858	0.924	0.978
M_lo_dt	f1	1942	0.73	0.755	0.816	0.84	0.859	0.922	1.0
M_lo_dn	f1	1942	0.734	0.75	0.815	0.839	0.859	0.923	1.0
M_lo_dtn	f1	1961	0.731	0.749	0.814	0.84	0.859	0.926	1.0
no_context	precision	1997	0.731	0.731	0.794	0.817	0.837	0.901	0.901
N_sh_1	precision	1997	0.75	0.75	0.801	0.821	0.84	0.896	0.901
N_sh_2	precision	1997	0.744	0.744	0.798	0.819	0.838	0.892	0.916

Continued on next page

Setup	Metric	Count	Min	Lower Whisker	Q1	Median	Q3	Upper Whisker	Max
N_sh_3	precision	1997	0.727	0.749	0.798	0.818	0.836	0.887	1.0
N_lo_1	precision	1994	0.733	0.738	0.799	0.821	0.84	0.9	1.0
N_lo_2	precision	1995	0.735	0.735	0.798	0.822	0.841	0.904	1.0
N_lo_3	precision	1996	0.739	0.739	0.798	0.821	0.841	0.901	1.0
M_sh_d	precision	1978	0.675	0.741	0.797	0.817	0.837	0.894	1.0
M_sh_dt	precision	1988	0.734	0.743	0.796	0.818	0.836	0.887	1.0
M_sh_dn	precision	1988	0.739	0.743	0.798	0.819	0.837	0.894	0.961
M_sh_dtn	precision	1990	0.702	0.745	0.798	0.819	0.837	0.893	0.905
M_lo_d	precision	1829	0.696	0.734	0.797	0.821	0.84	0.903	0.97
M_lo_dt	precision	1942	0.735	0.741	0.799	0.822	0.84	0.901	1.0
M_lo_dn	precision	1942	0.738	0.738	0.798	0.821	0.841	0.904	1.0
M_lo_dtn	precision	1961	0.695	0.747	0.799	0.821	0.84	0.896	1.0
no_context	recall	1997	0.721	0.763	0.836	0.863	0.885	0.957	0.987
N_sh_1	recall	1997	0.703	0.754	0.828	0.853	0.877	0.948	0.984
N_sh_2	recall	1997	0.701	0.75	0.827	0.854	0.879	0.956	0.984
N_sh_3	recall	1997	0.703	0.754	0.828	0.853	0.878	0.946	1.0
N_lo_1	recall	1994	0.714	0.753	0.827	0.855	0.879	0.953	1.0
N_lo_2	recall	1995	0.712	0.753	0.828	0.856	0.88	0.955	1.0
N_lo_3	recall	1996	0.7	0.75	0.828	0.855	0.881	0.958	1.0
M_sh_d	recall	1978	0.702	0.759	0.832	0.857	0.88	0.953	1.0
M_sh_dt	recall	1988	0.698	0.756	0.83	0.857	0.88	0.952	1.0
M_sh_dn	recall	1988	0.7	0.753	0.829	0.857	0.88	0.955	0.988
M_sh_dtn	recall	1990	0.701	0.757	0.83	0.857	0.879	0.95	0.983
M_lo_d	recall	1829	0.706	0.752	0.83	0.858	0.882	0.958	0.987
M_lo_dt	recall	1942	0.698	0.755	0.831	0.859	0.883	0.959	1.0
M_lo_dn	recall	1942	0.701	0.754	0.83	0.858	0.881	0.958	1.0
M_lo_dtn	recall	1961	0.699	0.753	0.83	0.858	0.882	0.96	1.0

References

Ali Mohamed Nabil Allam and Mohamed Hassan Haggag. The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, 2(3), 2012.

Gabriel Y. Arteaga, Thomas B. Schön, and Nicolas Pielawski. Hallucination detection in llms: Fast and memory-efficient finetuned models. *arXiv preprint arXiv:2409.02976*, 2024.

AWS. Amazon bedrock, 2024. URL `https://aws.amazon.com/bedrock/`. Accessed: 2025-05-26.

Masha Belyi, Robert Friel, Shuai Shao, and Atindriyo Sanyal. Luna: An evaluation foundation model to catch language model hallucinations with high accuracy and low cost. *arXiv preprint arXiv:2406.00975*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. Inside: Llms' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744*, 2024.

Yuyan Chen, Qiang Fu, Yichen Yuan, Zhihao Wen, Ge Fan, Dayiheng Liu, Dongmei Zhang, Zhixu Li, and Yanghua Xiao. Hallucination detection: Robustly discerning reliable answers in large language models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 245–255, 2023. URL `https://doi.org/10.1145/3583780.3614905`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, 2024. URL `https://aclanthology.org/2024.eacl-demo.16`.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Philip Feldman, James R Foulds, and Shimei Pan. Ragged edges: The double-edged sword of retrieval-augmented chatbots. *CoRR*, 2024.

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. 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, 2023. URL `https://aclanthology.org/2023.acl-long.910`.

Jeff Gennari, Shing-hon Lau, Samuel Perl, Joel Parish, and Girish Sastry. Considerations for evaluating large language models for cybersecurity tasks, 2024.

Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

Songyue Han, Mingyu Wang, Jialong Zhang, Dongdong Li, and Junhong Duan. A review of large language models: Fundamental architectures, key technological evolutions, interdisciplinary technologies integration, optimization and compression techniques, applications, and challenges. *Electronics*, 13(24):5040, 2024.

Xingyun Hong, Yan Shao, Zhilin Wang, Manni Duan, and Xiongnan Jin. Towards building a robust knowledge intensive question answering model with large language models. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 228–242. Springer, 2024.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. Q2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering. *arXiv preprint arXiv:2104.08202*, 2021.

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

Zhengjie Ji, Edward Choi, and Peng Gao. A knowledge base question answering system for cyber threat knowledge acquisition. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 3158–3161, 2022.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*, 2023.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*, 2024.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*, 2019.

Zefang Liu. Secqa: A concise question-answering dataset for evaluating large language models in computer security. *arXiv preprint arXiv:2312.15838*, 2023.

Johannes F Loevenich, Erik Adler, Remi Mercier, Alexander Velazquez, and Roberto Rigolin F Lopes. Design of an autonomous cyber defence agent using hybrid ai models. In *2024 International Conference on Military Communication and Information Systems (ICMCIS)*, pages 1–10. IEEE, 2024.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

Peter Markstein. The new ieee-754 standard for floating point arithmetic. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2008.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language :odels: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

MITRE. attack-stix-data. https://github.com/mitre-attack/attack-stix-data, 2024. Accessed: 2025-05-27.

W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

National Institute of Standards and Technology. Federal laboratory interagency technology transfer summary reports. Technical report, U.S. Department of Commerce, 2022. URL https://www.nist.gov/tpo/federal-laboratory-interagency-technology-transfer-summary-reports. Accessed: 2025-05-26.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing 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.

David Rau, Shuai Wang, Hervé Déjean, and Stéphane Clinchant. Context embeddings for efficient answer generation in rag. *arXiv preprint arXiv:2407.09252*, 2024.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. Ares: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476*, 2024.

Daniel Schatz, Rabih Bashroush, and Julie Wall. Towards a more representative definition of cyber security. *Journal of Digital Forensics, Security and Law*, 12(2):8, 2017.

Samaneh Shafee, Alysson Bessani, and Pedro M Ferreira. Evaluation of llm-based chatbots for osint-based cyber threat awareness. *Expert Systems with Applications*, page 125509, 2024.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*, 2021.

Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. Rethinking interpretability in the era of large language models. *arXiv preprint arXiv:2402.01761*, 2024.

Ben Snyder, Marius Moisescu, and Muhammad Bilal Zafar. On early detection of hallucinations in factual question answering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2721–2732, 2024.

Blake E Strom, Andy Applebaum, Doug P Miller, Kathryn C Nickels, Adam G Pennington, and Cody B Thomas. Mitre att&ck: Design and philosophy. In *Technical report*. The MITRE Corporation, 2018.

Norbert Tihanyi, Mohamed Amine Ferrag, Ridhi Jain, Tamas Bisztray, and Merouane Debbah. Cybermetric: A benchmark dataset based on retrieval-augmented generation for evaluating llms in cybersecurity knowledge. In *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 296–302, 2024.

S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Jiaheng Wei, Yuanshun Yao, Jean-Francois Ton, Hongyi Guo, Andrew Estornell, and Yang Liu. Measuring and reducing llm hallucination without gold-standard answers. *arXiv preprint arXiv:2402.10412*, 2024.

Xuansheng Wu, Haiyan Zhao, Yaochen Zhu, Yucheng Shi, Fan Yang, Tianming Liu, Xiaoming Zhai, Wenlin Yao, Jundong Li, Mengnan Du, and Ninghao Liu. Usable xai: 10 strategies towards exploiting explainability in the llm era. *arXiv preprint arXiv:2403.08946*, 2024.

Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Cheng Niu, Randy Zhong, Juntong Song, and Tong Zhang. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. *arXiv preprint arXiv:2401.00396*, 2023.

Maxime Würsch, Andrei Kucharavy, Dimitri Percia David, and Alain Mermoud. Llm-based entity extraction is not for cybersecurity. *Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE2023) and AI+ Informetrics (AII2023)*, 3451:26–32, 2023.

Weijia Xu, Sweta Agrawal, Eleftheria Briakou, Marianna J Martindale, and Marine Carpuat. Understanding and detecting hallucinations in neural machine translation via model introspection. *Transactions of the Association for Computational Linguistics*, 11:546–564, 2023.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211, 2024.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*, 2023.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. Retrieve anything to augment large language models. *arXiv preprint arXiv:2310.07554*, 2023a.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

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

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.