Automated Question-Answer Generation for Evaluating RAG-based Chatbots

Juan José González Torres, Mihai-Bogdan Bîndilă, Sebastiaan Hofstee, Daniel Szondy, Quang-Hung Nguyen, Shenghui Wang, Gwenn Englebienne
University of Twente
Drienerlolaan 5, 7522 NB Enschede
{j.j.gonzaleztorres, m.bindila, s.b.h.c.hofstee, d.g.szondy, nguyenquanghung}@student.utwente.nl
{shenghui.wang, g.englebienne}@utwente.nl

Abstract

In this research, we propose a framework to generate human-like question-answer pairs with long or factoid answers automatically and, based on them, automatically evaluate the quality of Retrieval-Augmented Generation (RAG). Our framework can also create datasets that assess hallucination levels of Large Language Models (LLMs) by simulating unanswerable questions. We then apply the framework to create a dataset of question-answer (QA) pairs based on more than 1,000 leaflets about the medical and administrative procedures of a hospital. The dataset was evaluated by hospital specialists, who confirmed that more than 50% of the QA pairs are applicable. Finally, we show that our framework can be used to evaluate LLM performance by using Llama-2-13B fine-tuned in Dutch (Vanroy, 2023) with the generated dataset, and show the method's use in testing models with regard to answering unanswerable and factoid questions appears promising.

Keywords: LLMs, Retrieval Augmented Generation, Chatbot Evaluation, Hallucination Detection

1. Introduction

Chatbots' performance has been greatly enhanced with recent advancements in RAG-based LLMs, where questions are supported by verified sources of information so that LLMs can answer consistently and accurately. However, evaluating these chatbots requires an enormous amount of labelled data that is often costly to produce in terms of human and financial resources. Moreover, evaluation datasets need to satisfy a few different criteria:

- **Covers large knowledge base:** A RAG pipeline often includes thousands of documents in various topics and language levels (scientific, conversational, etc.)
- **Includes different types of answers:** Answers can be factoid or long-form, depending on the type and format of questions. There might also be unanswerable questions, whether due to failure in the retriever or lack of pre-trained knowledge.

Therefore, a proper evaluation workflow should assess the chatbot's knowledge across all the topics covered in these documents and include all types of questions users could ask. In this research, we focus on automating the creation of a comprehensive QA dataset that satisfies the criteria above, which leads to the following research questions:

1. How to cover all topics when creating the questions?
2. How to account for questions that cannot be answered?
3. How to automatically generate and filter question-answer pairs starting from a set of documents?
4. To what extent can we compare LLMs' performance using generated data?

2. Related Work

The Question Generation (QG) branch of Natural Language Processing (NLP) has been of great interest recently due to the rising need for datasets for chatbot evaluation.

Usually, these datasets are produced based on a set of documents. In (Cohen et al., 2023), the authors use the QA pairs to form a knowledge base using the dataset Probably-Asked Questions (Lewis et al., 2021). Thus, QG can be perceived as a way of augmenting data for QA systems. The dataset mentioned before was also automatically generated for the task of Open-Domain Question Answering. It comprises 65 million questions in a four-step process composed of passage selection, span answer extraction based on named entities, question generation, and filtering (Lewis et al., 2021). The drawback of this method was that the generated answers were very brief. The solution implemented by the authors for this issue was to model the problem as a Long-Form Question-Answering one that creates open-ended questions that require explanation (Fan et al., 2019). The authors proposed a query-based multi-document summarization approach with sequence-to-sequence models.

Regarding the different types of questions, it is also possible to create them considering more sophisticated processes than reading comprehension.

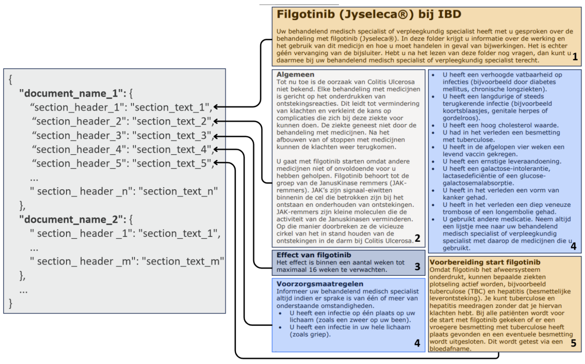
20 May, 2024.

As presented in (Wang, 2022), questions may be generated employing reasoning processes such as common sense, finding the most logical continuation of a sentence, or using deduction and induction given some premises to reach the correct conclusion. On the other hand, various techniques have also been proposed to remove irrelevant questions, such as n-gram similarity between question and context or scores given by another LLM to the quality of produced data (Yuan et al., 2022).

Detecting topics in texts is also one of the tasks researchers have covered the most in NLP. Some of them use Self-Organizing maps along with the k-means algorithm (de Miranda et al., 2020), others have focused on using graph nets for analyzing text embeddings (Romanova, 2021), and also the exploitation of temporal correlation in social media posts has been utilized to detect topics (Comito et al., 2019). Existing research tends to concentrate on clustering, but there are additional steps needed to transform raw text into meaningful topics.

3. Data

We perform experiments on a dataset of 1,320 leaflets (3,958 pages). These leaflets contain information in Dutch about different medical and administrative procedures to help patients navigate hospital services and find medical information. Figure 1 illustrates an example of one leaflet's page containing multiple sections. A section has a heading and one or more paragraphs that can include bullet points, tables, and images and can span across multiple pages or columns.



```
{
  "document_name_1": {
    “section_header_1": "section_text_1",
    “section_header_2": "section_text_2",
    “section_header_3": "section_text_3",
    “section_header_4": "section_text_4",
    “section_header_5": "section_text_5",
    ...
    " section_ header _n": "section_text_n"
  },
  "document_name_2": {
    " section_ header _1": "section_text_1",
    ...
    " section_ header _m": "section_text_m"
  },
  ...
}
```

Filgotinib (Jyseleca®) bij IBD

Uw behandelend medisch specialist of verpleegkundig specialist heeft met u gesproken over de behandeling met filgotinib (Jyseleca®). In deze folder krijgt u informatie over de werking en het gebruik van dit medicijn en hoe u moet handelen in geval van bijwerkingen. Het is echter géén vervanging van de bijsluiter. Hebt u na het lezen van deze folder nog vragen, dan kunt u daarmee bij uw behandelend medisch specialist of verpleegkundig specialist terecht. 1

Algemeen
Tot nu toe is de oorzaak van Colitis Ulcerosa niet bekend. Elke behandeling met medicijnen is gericht op het onderdrukken van ontstekingsreacties. Dit leidt tot vermindering van klachten en verkleint de kans op complicaties die zich bij deze ziekte voor kunnen doen. De ziekte geneest niet door de behandeling met medicijnen. Na het afbouwen van of stoppen met medicijnen kunnen de klachten weer terugkomen.

U gaat met filgotinib starten omdat andere medicijnen niet of onvoldoende voor u hebben geholpen. Filgotinib behoort tot de groep van de JanusKinase remmers (JAK-remmers). JAK's zijn signaal-eiwitten binnenin de cel die betrokken zijn bij het ontstaan en onderhouden van ontstekingen. JAK-remmers zijn kleine moleculen die de activiteit van de Januskinasen verminderen. Op die manier doorbreken ze de vicieuze cirkel van het in stand houden van de ontstekingen in de darm bij Colitis Ulcerosa. 2

Effect van filgotinib
Het effect is binnen een aantal weken tot maximaal 16 weken te verwachten. 3

Voorzorgsmaatregelen
Informeer uw behandelend medisch specialist altijd indien er sprake is van één of meer van onderstaande omstandigheden.

- U heeft een infectie op één plaats op uw lichaam (zoals een zweer op uw been).
- U heeft een infectie in uw hele lichaam (zoals griep).
- U heeft een verhoogde vatbaarheid op infecties (bijvoorbeeld door diabetes mellitus, chronische longziekten).
- U heeft een langdurige of steeds terugkerende infectie (bijvoorbeeld koortsblaasjes, genitale herpes of gordelroos).
- U heeft een hoog cholesterol waarde.
- U had in het verleden een besmetting met tuberculose.
- U heeft in de afgelopen vier weken een levend vaccin gekregen.
- U heeft een ernstige leveraandoening.
- U heeft een galactose-intolerantie, lactasedeficiëntie of een glucose-galactosemalabsorptie.
- U heeft in het verleden een vorm van kanker gehad.
- U heeft in het verleden een diep veneuze trombose of een longembolie gehad.
- U gebruikt andere medicatie. Neem altijd een lijstje mee naar uw behandelend medisch specialist of verpleegkundig specialist met daarop de medicijnen die u gebruikt. 4

Voorbereiding start filgotinib
Omdat filgotinib het afweersysteem onderdrukt, kunnen bepaalde ziekten plotseling actief worden, bijvoorbeeld tuberculose (TBC) en hepatitis (besmettelijke leverontsteking). Je kunt tuberculose en hepatitis meedragen zonder dat je hiervan klachten hebt. Bij alle patiënten wordt voor de start met filgotinib gekeken of er een vroegere besmetting met tuberculose heeft plaats gevonden en een eventuele besmetting wordt uitgesloten. Dit wordt getest via een bloedafname. 5

Figure 1: Format of a leaflet page

4. Method

We propose a multi-step framework to automate the creation process of the QA dataset, which is shown in Figure 2. In the following sections, we will discuss in detail each step.

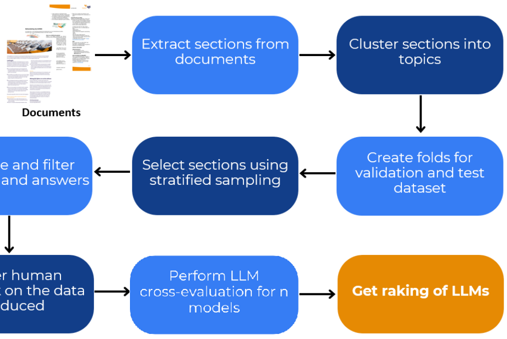

Documents
Extract sections from documents
Cluster sections into topics
Create folds for validation and test dataset
Select sections using stratified sampling
Generate and filter questions and answers
Gather human feedback on the data produced
Perform LLM cross-evaluation for n models
Get raking of LLMs

Figure 2: Steps of the proposed framework

4.1. Section Extraction and Grouping

To ensure the uniqueness of the QA pairs, we create targeted questions based on sections. Questions generated on each section instead of each page or document are easier to evaluate automatically (Yuan et al., 2022). To extract these sections, we developed a PDF parser that considers font characteristics such as size and style together with the text structure and spacing between paragraphs. The sections extracted incorrectly are labeled as anomalies by BERTopic in a downstream topic detection phase and are not considered afterward. With our parser, we extracted 13,216 sections out of 1,320 digital leaflets, which are machine-readable. The algorithm is presented in Appendix 8.

The size of chunks retrieved during RAG directly influences the output quality. Rather than using sections during RAG, we introduce the concept of "emulated pages", obtained by sequentially grouping document sections. This approximates the size of the original pages while avoiding the division of sections, resulting in a median of 842 tokens and a slight deviation from the general recommended 1000-token chunk size (Rameel Ahmad, 2024).

4.2. Formation of Topics

After that, we create questions based on the extracted sections. The leaflets are clustered into groups based on the topics they covered, using BERTopic (Grootendorst, 2022) - a modelling framework that extracts interpretable and concise topics. To ensure BERTopic's performance, two main hyperparameters must be considered: embedding's dimensionality and minimum cluster size.

To evaluate the quality of the clusters for each dimensionality of the embeddings, three groups of metrics are used:

- **Geometric**: Silhouette score (Davies and Bouldin, 1979a), Calinski-Harabasz (Caliński and Harabasz, 1974), and Davies-Bouldin (Davies and Bouldin, 1979b) indices.
- **Robustness**: Another interesting way to as-

sess the quality of clusters is adding noise to the data (Davidson et al., 2001), in this case, to the embeddings. Then, the higher the number of embeddings clustered into the same group with and without noise, the more robust the clusters are.

- **Document-cluster evaluation**: If we assume that only one topic is treated per document, a reasonable metric for the clusters is that all sections of a given document should ideally belong to the same cluster (except for certain exceptions, such as contact sections). If it is known that the documents contain information about different topics, this metric should be ignored. If that is not the case, the higher the matching, the better the clustering. The expression 2 yields this metric, with d_i as the metric for each document, s_j the number of sections belonging to topic j, N the number of sections in the document, and M the number of documents in the cluster.

$$d_i = \max_j(\frac{s_j}{N}) * 100 \tag{1}$$

$$DC = \frac{\sum_{i=1}^{n} d_i}{M} \tag{2}$$

4.3. Cross-evaluation

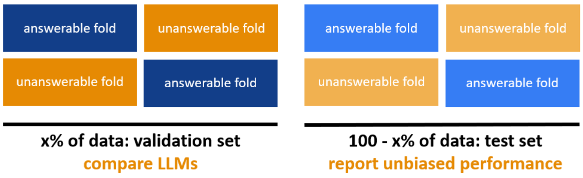


Figure 3: Cross-evaluation setup

We designed a Cross-Evaluation (CE) method for RAG-based LLMs inspired by classical machine learning techniques. This method accounts for unanswerable questions without external knowledge bases by grouping the emulated pages into disjoint information groups ("folds"). When faced with an unanswerable question, the model responds with "I don't know" instead of inventing an answer, allowing us to evaluate its hallucination level.

Figure 3 illustrates the CE setup that contains the validation and test sets. Each has two folds with a modifiable ratio of test-validation test. We use the first set to compare the performance of LLMs based on the metrics described in section 4.7, and then the second set to report the unbiased performance of the best model.

Our CE setup involves two iterations for validation and two for testing. The folds are utilized in each iteration as depicted in Figure 4. Specifically, all emulated pages from the answerable fold are stored in the vector database (e.g., Qdrant (Qdrant, 2024)), and questions are posed based on both answerable and unanswerable folds. Subsequently, the metrics presented in section 4.7 are measured for each iteration and averaged for each set, providing a comprehensive evaluation of the model's performance.

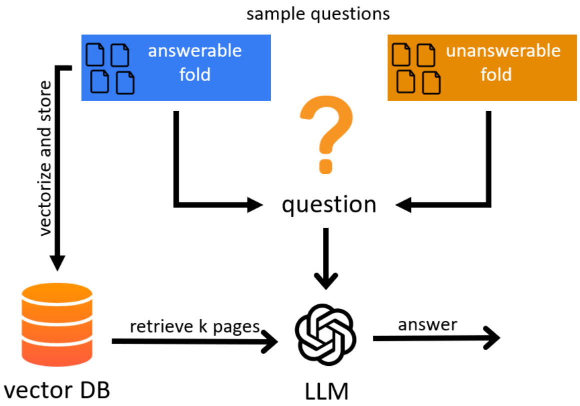


Figure 4: Use of folds while answering questions

4.4. Creation of Folds

As presented in the section 4.2, the folds are created based on the discovered topics. We use the sections from these topics in a vectorized form represented by low-dimensionality embeddings created by BERTopic. We designed a bottom-up hierarchical approach to group the topics into folds. This grouping enables us to use stratified sampling based on topics within each fold to make the selected sections as diverse as possible. Our algorithm minimizes the probability of having overlapping information in any pair of folds by maximizing the folds' distance in the embedding space and removing the sections from common pages or documents that are in different folds.

Even though folds are represented in the CE setup by groups of emulated chunks stored in the vector database, we create the folds with sections as the atomic unit. Hence, these folds can be viewed from two perspectives:

- groups of sections from which we sample and create questions
- groups of emulated chunks used during CE that contain the sections

We choose to have two folds since we are only interested in minimizing the probability of them having overlapping information. Moreover, adding more folds implicitly reduces the distance between them, and two folds are sufficient to simulate unanswerable questions.

4.4.1. Algorithm to Create the Folds

The algorithm is highly customizable, having the following parameters: ratio of folds, test set percentage, number of sampled sections for creating questions, and whether the folds should contain sections from different pages or documents. As shown in Figure 5, the procedure includes a series of steps that will be described below.

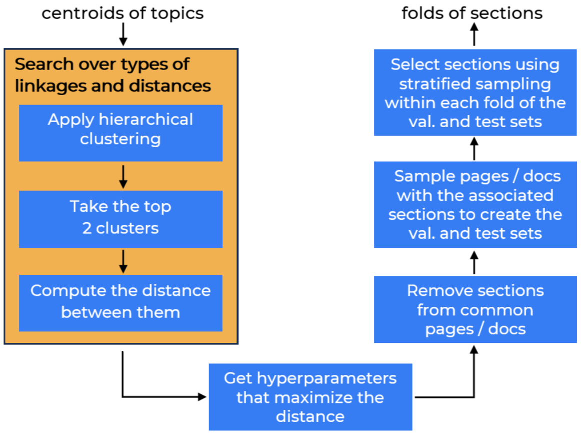

centroids of topics
folds of sections
Search over types of linkages and distances
Apply hierarchical clustering
Take the top 2 clusters
Compute the distance between them
Get hyperparameters that maximize the distance
Remove sections from common pages / docs
Sample pages / docs with the associated sections to create the val. and test sets
Select sections using stratified sampling within each fold of the val. and test sets

Figure 5: Algorithm for creating folds

Since topics are represented by compact clusters without outliers, as presented in section 5.2, we clustered their centroids using the agglomerative approach of the hierarchical clustering (Müllner, 2011). This approach is optimal since we can use various linkage criteria between clusters and working only with the centroids makes the computation extremely efficient. The distance between folds is maximized by searching over the space defined by the distance type (Euclidean or cosine) and linkage criteria (single, complete, average, or ward).

After clustering with each set of hyperparameters, the newly created clusters are evaluated by computing the average of the single and average linkage using the Euclidean distance. We chose this distance to account for the space between all points and, simultaneously, to weigh the distance between the closest points more.

In the next step, sections that appear in pages or documents with sections in both folds are removed to avoid having common information in both folds during the cross-evaluation. Next, we sample pages or documents and their sections at the fold level to create validation and test datasets of sections following the set ratio. This step returns two folds for validation and two for test sets.

Finally, we employ stratified sampling within each fold to select a set of diverse sections covering all topics based on which questions are created. In our setup, strata are groups of sections from the same topic.

4.5. Types of Questions

For our purpose, we categorized the questions based on two different criteria: if the question is answerable based on the leaflets, and if the question is factoid or long-form.

Long-form, open-ended questions assess the machine's ability to provide helpful advice based on its database since the chatbot is expected to interpret and explain information relevant to patient inquiries. (e.g., "What is actigraphy and how can it diagnose sleep problems in children?"). Meanwhile, factoid questions test the chatbot's ability to accurately retrieve facts (e.g., phone numbers or email addresses). They verify if the LLM can locate precise information without hallucinating and are evaluated by a pass/fail metric. (e.g., "What is the telephone number of the radiology department?")

4.6. Q&A Generation and Filtering

A two-stage approach is utilized to create the question set for long-form and factoid questions: First, a larger, diverse set of questions is generated in Dutch using GPT-3.5-turbo-instruct (OpenAI, 2022). Then, we filter the questions using embedding similarity, ROUGE score (Lin, 2004) between the answer and the section, and sorted by a score assigned by the model based on examples. Top questions are selected per source section to retain the distribution of QAs concerning the data in the folds. We present the details of this algorithm in Appendix 8.

For phone numbers and email addresses from leaflets, we use the same two-stage mentioned above to generate a question for each section and the entity extracted from it, with a modification: rather than assessing section-answer similarity, we directly verify the presence and accuracy of the email/phone number in both the answer and the section.

In the first stage, 5000 QA pairs are generated. The filtering steps reduce this to 500 pairs, with each final question corresponding to one sample section to avoid distorting the previous distribution of samples.

4.7. Measured Metrics

Initially, hospital specialists will perform a qualitative evaluation of the generated QA dataset. Only the QA pairs labelled as correct are used in the cross-evaluation procedure to test various LLMs.

Secondly, answers should be analyzed quantitatively. We compute the hallucination rate for every answer - the percentage of unanswerable questions that would have been answered without the proper information. However, these events could not only be due to hallucination but also to a poor division in the folds. Therefore, we measure the percentage of answers where the correct information has been given to evaluate factoid questions. For long-form questions, we use the standard metric for the long-answer questions: BLEU, ROUGE-1, ROUGE-2, ROUGE-L, and BLEURT (Celikyilmaz et al., 2020). Finally, humans evaluate the correctness of an answer.

4.8. Using Folds to Evaluate Different Models

A Dutch language model (given the data is in Dutch), Llama-2-13B-Dutch with 8-bit quantization (Vanroy, 2023) is human-evaluated to determine how the method, and specifically the different folds, can be used in order to determine how well certain LLMs perform. These are separated into three distinct classes:

- The performance on long-form answerable questions (folds 1 and 2). This performance is the percentage of correctly answered questions evaluated by a human.
- The performance on long-form unanswerable questions (with disjoint folds A and B, the context of fold A is used for questions from fold B and vice versa). In this way, our method can be used to determine to what extent a model can indicate that there is no answer to the question based on the given context. This metric is defined as the percentage of questions answered by the model with an answer that makes it clear that the model does not have enough information to answer the question evaluated by a human. Seventy-eight annotations are made for both combinations of disjoint folds.
- The performance on answerable factoid questions. A fold F containing only factoid questions can be used to determine whether a model can extract said factoids well. This performance is defined as the percentage of questions that are answered by the model with an answer that correctly extracts the factoid information evaluated by a human. Five annotations were made, given that five factoid questions were present in the dataset.

5. Results

5.1. Formation of Topics

The performance of BERTopic with different values for the hyperparameters "min_cluster_size" of the HDBSCAN and "n_components" of the UMAP algorithm is compared: $5, 10, 15, 20, 25, 30, 35, 40, 45, 50$.

We set the number of topics to 200 a priori, considering that the number of leaflets and too many clusters would result in too numerous topics. A higher embedding dimensionality implies a larger accuracy of the clustering algorithm, but less information covering the section's content will be kept. Therefore, dimensional values under 10 are hardly acceptable, so a trade-off solution should be chosen. All this a priori knowledge goes along with the results of the metrics, which are enumerated in the following list:

- **Geometry metrics**: Depends on the minimum cluster size: better results for small values (up to 35). There is no clear dependence on the embedding dimensionality.
- **Robustness metrics**: Better for intermediate values of minimum cluster size (between 20 and 30) and highly depend on the dimensionality of the embedding, where high dimensions (over 15) result in lower robustness. Here, results for high values of both variables are deceitfully good only due to the lower quantity of clusters. The same behavior is observed in the document-cluster matching score.
- **Document-cluster metrics**: Highly dependent on the minimum cluster size (which was predictable). It is better to have a more increased value, up to the extreme case of too few clusters—no apparent dependence on embedding dimensionality.

The interest lies in exploring different values for hyperparameters, leading to the need for a trade-off solution. While results on semantic coherence suggest favoring fewer topics, geometry-based calculations indicate that more topics would better represent document information. Ultimately, we decided to increase the embedding dimensions from the default 5 to 15. Similarly, the "min_cluster_size" will be adjusted from the default 10 to 30 elements. The number of clusters typically hovers around 70 topics, which will be discussed in the following section. Outliers are not a significant concern since they remain within acceptable proportions, as demonstrated in the implementation example of BERTopic. Additionally, the abundance of labelled sections mitigates concerns about outliers.

These chosen hyperparameters balance geometric robustness metrics and the clustering of sections within the same document, ensuring effective representation.

5.2. Topic Analysis

This section analyzes two sets of topics created with BERTopic. The main difference between them is the transformer that creates each section's embeddings. The first one is built using distiluse-base-multilingual-cased-v1 (DBMC-v1), a distilled version of the model presented in (Yang et al., 2019). The second one is based on paraphrase-multilingual-MiniLM-L12-v2 (PMM-L12-

v2), a multilingual version of paraphrase-MiniLM-L12-v2 (Reimers and Gurevych, 2019).

These topics are composed of sections represented as embeddings with 15 dimensions. Their attributes are described in Table 1.

statistic	DBMC-v1	PMM-L12-v2
min topic size	32	30
median topic size	66	79
max topic size	634	1063
topics no	71	62
outliers %	40.55%	32.17%

Table 1: Statistics of the generated topic clusters

Many sections are detected as outliers and not included in any topics, leading to very compact clusters represented well in space by their centroids.

Lastly, the data does not reveal any correlation (-0.032 or -0.051, depending on the set of topics) between cluster size and the minimum distance from any cluster to the closest one. This indicates that topics are not isolated based on size; they are all positioned randomly in space.

5.3. Fold Analysis

As we have a parameterizable algorithm, we chose a test set percentage of 20% and specified that the two folds should contain sections from different documents. After running the procedure presented in section 4.4 with each set of topics created with BERTopic, we found folds with enough sections to build the validation and test sets only in the case of four combinations of hyperparameters.

As shown in Table 2, the best results are obtained for both sets of topics when the ward linkage criterion is used together with the Euclidean distance. The space between the folds increased between 6.7 and 8.21 times compared to the initial distance between the topics.

set of topics	DBMC-v1	DBMC-v1	DBMC-v1	PMM-L12-v2
linkage criterion	ward	complete	complete	ward
distance type	Euclidean	cosine	Euclidean	Euclidean
min dist topics	0.2783	0.2783	0.2783	0.2481
min dist folds	2.2882	1.923	1.8662	1.9548
small fold ratio	0.1175	0.3741	0.1574	0.4727
avg folds per doc	1.2104	1.3285	1.1208	1.3399
valid sections %	72.12%	55.84%	84.22%	55.28%

Table 2: Results of the hierarchical clustering

Our approach is better than forming a fold based on the most isolated clusters because we create them based on more topics. Regarding fold ratio, the folds are almost even in the case of PMM-L12-v2. The last two rows of Table 2 refer to the number of valid sections from the perspective that the folds contain sections from different documents. A larger average fold per document implies that more sections must be filtered out. The remaining sections are used to compose the 80-20 validation-test split, followed by the stratified sampling step.

These sampled sections are the basis for creating questions rated by professionals. A large enough and agreed-upon size to assess was 500. The sections are selected concerning the folds ratio rounded to the first decimal, meaning that for PMM-L12-v2, we have a ratio of 0.5. This leads to the next sizes: validation fold 1 - 200, validation fold 2- 200, test fold 1- 50, and test fold 2 - 50.

To reduce human effort and focus on the higher-quality set of questions, we will only use the sampled sections in the case of PMM-L12-v2. In the other scenario, even though the distance between the folds is larger, the second fold is too small, resulting in similar sections that do not cover the entire scope of the information.

The final evaluation of these folds will be performed during CE. That is the final check if any overlapping information between folds is present.

5.4. Human Evaluation

Five hospital specialists annotated these 490 questions, the generated answers, and the reference section. For each QA pair, the annotators were told to choose one or more options: irrelevant question, too specific question, wrong answer, incomplete or ambiguous answer, correct answer, and, optionally, to write a short feedback. Out of the total questions, 85 were double-annotated for quality control, one wasn't evaluated, and the remaining were reviewed by a single random annotator from the pool of five. The annotators jointly agreed in 64.71% of cases and partially agreed in 15.29% of cases, meaning that they picked multiple options, of which at least one is the same.

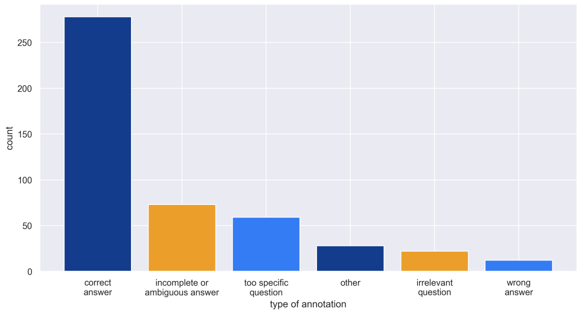


Figure 6: Distribution of annotations

Figure 6 shows the distribution of annotations for all questions except the 17 questions that produced a disagreement. A total of 278 questions were labelled as having correct answers. In 28 cases named "other", the annotators did not pick any predefined option but wrote a comment. Mostly, comments refer to the fact that the question or answer is too specific, the answer is incomplete, the question formulation is strange, or the answer tone is offensive in only one instance. Out of the reviewed set, in only 12 cases, a "wrong answer" was generated. This low number validates the quality of the generated data.

Based on this feedback, we filtered the questions to use only the ones labelled as having correct answers in the cross-evaluation procedure.

5.5. Cross Evaluation Results

We used the dataset for cross-evaluation on the GPT-3.5 Turbo model using 209 question-answer pairs from two training folds, 76 from fold one and 133 from fold two. The evaluation assesses the model's information retrieval capability and response appropriateness along various metrics, which are presented in Table 3.

Metric	Fold One	Fold Two
Using Fold One RAG (%)	57.33	7.52
Using Fold Two RAG (%)	12.00	62.41
BLEU Score	0.0734	0.0777
ROUGE-1 Score	0.312	0.261
ROUGE-2 Score	0.188	0.154
ROUGE-L Score	0.259	0.222
BLEURT Score	0.592	0.517
Facts Correctly Extracted (%)	20	-

Table 3: Cross-evaluation results for GPT-3.5 Turbo

5.5.1. Evaluation on Llama-2-13B-Dutch

The Llama-2-13B-Dutch model has been evaluated using the folds to determine its performance on questions it cannot answer. We consider a valid reply if the model states that there is not enough information to answer. Furthermore, the factoid fold has been used to determine how well the model can extract data from a context containing specific factual information. These results can be seen in Table 4, where Q1_CTX2 represents fold 1 with context from disjoint fold 2, Q2_CTX1 represents fold two with context from disjoint fold one, and the Factoids Corr. % represents the percentage of factoids correctly extracted by the LLM.

Q1_CTX2 Corr. % (n=78)	Q2_CTX1 Corr. % (n=78)	Factoids Corr. % (n=5)
0	0	100

Table 4: Evaluation of unanswerable and factoid questions

On the total 278 correct determined questions, which are answerable, the model was further tested. The results of the numerical evaluation are shown in Table 5.

5.5.2. Answering When the Corresponding RAG Pages are Loaded

To assess the system's ability to retrieve information correctly and to decline to answer in case no information is available, we did a cross-evaluation procedure presented in section 4.3, For this, we expect that if the corresponding fold is loaded, the model should try to answer all questions, while if the unrelated documents are used, it should not answer any of the questions.

The results are found in the first two rows of Table 3. Declines to answer were either hard-wired from the failure of the retrieval or manually labeled if the model declined to answer (even though the retrieval gave some unrelated results).

5.6. Human Evaluation

With the human evaluation done by medical specialists, we can deduce that automatic question and answer generation is a feasible way to create relevant questions, as around 83,46% of questions were considered by them as being relevant. Many cases were flagged as irrelevant or having too specific questions, which might require adjusting, but it also gives useful feedback on the level of specificity required for this type of chatbot. Having over 56.73% fully correct QA pairs means that a reasonable portion can be used directly for evaluating LLMs.

5.7. Cross-evaluation

5.7.1. Answerable and Unanswerable Questions

The results show that the chatbot is unwilling to answer around 40% of the answerable questions. While we might need to consider that it is affected by the style of questions we had, this leaves room for improvement in the system, most likely in the retrieval.

For cases when the data is not available, the chatbot correctly declines to answer around 90% of the time, which exceeded our expectations. Considering that, seemingly, in most of the cases, it was not due to the retrieval not giving any results, the LLM decided that it did not have enough information.

5.7.2. Performance Metrics

All performance metrics should be treated as a baseline for comparison with other models; on their own, they might not give a clear picture of the answer quality.

The results around 0.07 for BLEU are low, probably due to the LLM's tendency to rephrase the content, resulting in low N-gram overlap.

	BLEU Score	BERTScore Precision	BERTScore Recall	BERTScore F1	ROUGE-1 F-measure	ROUGE-L F-measure
	Results for tested models on verified dataset (n=278)					
Llama-2-13b-Dutch	17.9	0.761	0.833	0.793	0.466	0.417

Table 5: Llama2 results for answerable questions

The ROUGE scores vary between 0.15 and 0.3, with Rouge-2 scores being the lowest. The 0.3 score might seem acceptable, but it needs to be used as a comparative value.

In examining the BLEURT score, we recognize that it got the highest values, with its 0.5-0.6 ratings. As this metric is trained to better correlate with human judgment, having a satisfactory rating will give a better comparison later, once it can be compared with other variants of the chatbot.

5.7.3. Factoid Questions

In this case, the chatbot underperformed by not finding the correct address, although this is statistically insignificant since the number of factoid questions was low.

5.7.4. Evaluation on Llama-2-13B-Dutch

The evaluation of the Llama-2-13B-Dutch language model utilizes different folds to assess its performance on answerable, unanswerable, and factoid questions. Tables 2 and 5 demonstrate the effectiveness of this method in evaluating model performance across these aspects. Specifically, the fine-tuned Dutch model struggles to identify unanswerable questions when contextually lacking necessary information. However, it excels in extracting factoid details and performs well on answerable questions.

5.8. Limitations and Challenges

The proposed framework is limited; thus, the resulting folds can be unbalanced. The PDF parser that extracts the sections works only on a specific type of leaflets and should be extended to be more general. We have not found a way to ensure an approximate number of sections in each fold. Another issue is that it requires humans in the loop to evaluate the QA pairs used during cross-evaluation. Additionally, we create factoid questions only related to named entities such as phone numbers and email addresses, while many more facts are present in the leaflets. Additionally, regarding the framework's usability in assessing the performance of models in answering answerable, unanswerable, and factoid questions, an example containing two folds of 78 questions, answers, and context triples is quite limited. Furthermore, there were only five factoid questions in the respective fold. Despite these examples still showing the method's potential, said sample sizes are relatively small, and fold-specific evaluation has only been done for one model. Future work expanding the evaluation of this framework is therefore welcomed.

6. Conclusion

This research paper presented a framework for evaluating RAG-based chatbots from a set of documents by automatically generating QA pairs and employing a cross-evaluation procedure that accounts for unanswerable questions. Our method enables the comparison of LLMs using various metrics for assessing long-form and factoid questions. The human evaluation results highlight the quality of the produced QA pairs, with 83,46% relevant questions and only 2.44% wrong answers. Moreover, although there are various limitations, we successfully demonstrated that our framework can be used to evaluate LLMs such as Llama-2-Dutch-13B or GPT-3.5 Turbo with a dataset of hospital leaflets for patients. The project's source code and the created dataset are publicly available at this link.

7. Acknowledgement

We acknowledge the Ziekenhuis Groep Twente (Hospital Group Twente - ZGT), Hengelo, The Netherlands for providing us with the leaflets and the computing power, as well as for manually annotating the generated QA pairs.

8. Bibliographical References

Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.

William W Cohen, Wenhu Chen, Michiel De Jong, Nitish Gupta, Alessandro Presta, Pat Verga, and

John Wieting. 2023. Qa is the new kr: Question-answer pairs as knowledge bases. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15385–15392.

Carmela Comito, Agostino Forestiero, and Clara Pizzuti. 2019. Word embedding based clustering to detect topics in social media. In *IEEE/WIC/ACM International Conference on Web Intelligence*, WI '19, page 192–199, New York, NY, USA. Association for Computing Machinery.

George S Davidson, Brian N Wylie, and Kevin W Boyack. 2001. Cluster stability and the use of noise in interpretation of clustering. In *Information Visualization, IEEE Symposium on*, pages 23–23. IEEE Computer Society.

David L Davies and Donald W Bouldin. 1979a. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.

David L. Davies and Donald W. Bouldin. 1979b. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

Guilherme Raiol de Miranda, Rodrigo Pasti, and Leandro Nunes de Castro. 2020. Detecting topics in documents by clustering word vectors. In *Distributed Computing and Artificial Intelligence, 16th International Conference*, pages 235–243, Cham. Springer International Publishing.

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

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure.

LangChain, Inc. 2024. Output-fixing parser. `https://python.langchain.com/docs/modules/model_io/output_parsers/types/output_fixing`. Accessed: 2024-02-02.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. Paq: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Daniel Müllner. 2011. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*.

OpenAI. 2022. GPT-3.5 Turbo: Language Model. `https://platform.openai.com/docs/models/gpt-3-5-turbo`.

Qdrant. 2024. Qdrant vector database.

Syed Rameel Ahmad. 2024. Enhancing multilingual information retrieval in mixed human resources environments: A rag model implementation for multicultural enterprise. *arXiv e-prints*, pages arXiv–2401.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Alex Romanova. 2021. Detect text topics by semantics graphs. In *Proceedings of the 2nd International Conference on Blockchain and Internet of Things (BIoT 2021)*, volume 11.

Bram Vanroy. 2023. Language resources for dutch large language modelling.

Zhen Wang. 2022. Modern question answering datasets and benchmarks: A survey.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. 2019. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*.

Xingdi Yuan, Tong Wang, Yen-Hsiang Wang, Emery Fine, Rania Abdelghani, Pauline Lucas, Hélène Sauzéon, and Pierre-Yves Oudeyer. 2022. Selecting better samples from pre-trained llms: A case study on question generation. *arXiv preprint arXiv:2209.11000*.

Appendix A: PDF Parser for Section Extraction

The algorithm for parsing the PDF leaflets has the following steps:

1. set manually the area of interest that contains the text and excludes the header and footer with metadata such as page number and date

2. find the majority font size associated with each line of text

3. divide the PDF into groups of paragraphs, called sections, based on two delimiters: if the space between them is larger than average, or if a line of text written in a larger-than-average font is encountered

4. handles the case where entire paragraphs are written in a larger font, and each line is recognized as a separate section, merging them into a section without a header

5. merge sections that do not have a header with the previous section and header-only sections with the following sections

Our approach works even in edge cases, correctly separating sections that span multiple pages, contain bullet points, or have paragraphs delimited in various ways. The goal is to extract all sections while minimizing the number of detections as a section of a group of sections or a part of a section.

The parser was designed specifically for this dataset while testing it on 11 representative PDFs, including a comprehensive set of edge cases. It was then manually verified with a larger random samp

Appendix B: Generation of Question-Answer Pairs

B.1 Long-Form Questions

An instruction-tuned LLM was utilized to produce a wide range of naturally formed questions without much constraint on the type of questions created. However, the quality of the questions is highly dependent on the model's performance on these tasks. A factor that made the task more difficult was that the questions needed to be generated in Dutch. We used GPT 3.5 for this purpose, which does support Dutch.

For the language model to give us the needed text, we needed to create instructions that precisely explained the task. This was done by creating a custom prompt for the task and progressively improving it. The prompt was created in Dutch, as this seemed to cause the LLM to reliably continue using the language upon asking for completion. The prompt describes the main task and "domain" and includes the selected section.

Since we had a few examples of question-answer pairs, we used them to employ a few-shot prompting techniques. This helped the model find the right tone and length for the reference answers.

As another factor to increase the variance of the questions, we added a few random roles like "recovering patient", "elderly patient", or "parent of a sick child" so that the model creates more varied scenarios. During use, certain additions were made to the prompt to create more suitable questions.

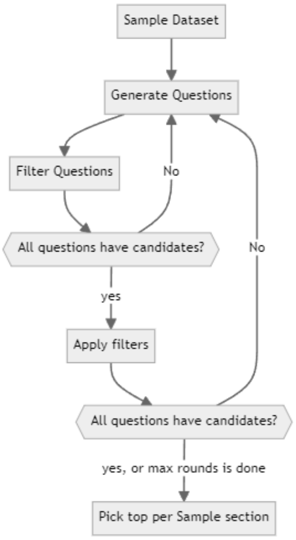

Sample Dataset
Generate Questions
Filter Questions
No
All questions have candidates?
No
yes
Apply filters
All questions have candidates?
yes, or max rounds is done
Pick top per Sample section

Figure 7: Long-form question generation workflow

To make the output machine parseable, we decided on a JSON scheme. The model struggled to follow these precisely enough, so the scheme description and instruction, but repeating the instructions at the beginning and end of the prompt, seemed to improve the rate of successful generation. Even with the changes, we still had several cases where the model failed to follow the scheme properly. As an additional step, "output fixing LLM"(LangChain, Inc., 2024) was employed to transform the faulty answers to the scheme. This was done since fixing the format was less costly than retrying the generation, and fixing the scheme was more reliable than the generation.

Filtering During the generation stage, we created more questions than we needed; however, most were not unique or high-quality enough.

We used cosine similarity on the question embeddings to filter out repeating or similar questions. Then, we executed a "drop out," where we discarded one for highly related questions until we reached the desired threshold.

We tried to ensure that the generated Q-A pair was related to the source. Unfortunately, there were quite a few cases where the model got "inspired" by the few-shot example and created content related to that over the context. We used ROUGE as a similarity metric with a low threshold on the generated answer and the context.

We needed to remove the questions that could've been considered "Short-form factoid" questions to avoid accidentally mixing the two types. To remove these, a basic rule was implemented. All questions that had a short answer and contained one of the Entity types we chose to extract were discarded.

B.2 Factoid Questions

We wanted to ask some highly targeted questions to evaluate the chatbot's ability to recite small information sections. This was necessitated because every time the model could not answer, the user might ask for a way to contact a human, which, in the hospital's case, would be the already existing contact phone number or address related to the topic. Many of these direct contacts were already included in the flyers we used.

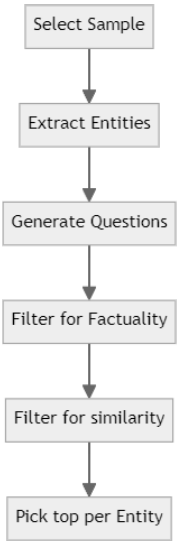


Figure 8: Factoid question generation workflow

Entity Extraction For extracting entities of interest, we experimented with some different options. Even though there are many Named Entity Recognition software, such as Flare, we decided to only include the most basic regex-based "phone number" and "email" extractions, as these types of data created direct questions that were suitable for our purpose.

Generation Similarly to the previously mentioned long-form generation, we used an LLM. The main difference was in the instruction. In addition to the context, the selected fact was provided, and all other details and examples were modified to fit the new format.

Filtering The questions were filtered by doing a back check on the answer, verifying that the original "fact" is extractable.

Similarly to the long-form questions, the factoid questions were filtered by their embeddings' cosine similarity.

The correctness of the extracted entities was verified at the end of this step and during the human evaluation step.