Final Project

Modern Information Retrieval - Fall-1403

Designed by: Narges Ghanei

Introduction

Nowadays, many people search for answers to their questions online, and healthcare-related questions are one of the most commonly searched topics. This makes it a crucial task to meet the increasing demand for reliable health information on the internet. As a result, many efforts are focused on processing these questions and finding the best answers. In the context of the healthcare domain, people often ask complex questions with excessive details that may not be necessary for answering the core query. Summarizing these questions helps focus on the key information, enabling users to get more direct and relevant answers. However, a key challenge is the lack of large, high-quality datasets for training models, particularly in specialized fields like healthcare, where expert knowledge is required for annotation. To address this, the project explores a new approach for creating diverse and useful healthcare question summaries using synthetic data, without the need for large datasets, in low-resource settings.

Project Overview

As mentioned, the solution to the problem of lacking high-quality datasets in the healthcare domain is to create a good and diverse dataset without needing expert oversight. Studies have shown that this can be achieved through a simple solution. We start with a small dataset of healthcare-related questions that people have searched for. The task is to apply round-trip translation (RTT), which involves translating questions to a different language and then back to English, generating different versions of the same question. This approach helps to create a larger dataset for training and summarization tasks. However, not all the generated questions are useful, as they may contain repetitive information. To address this, we use three different techniques—Frechét Question Distance (FQD), Precision Recall Question Distance (PRQD), and Question Semantic Volume (QSV)—to select the most informative and useful questions from the new subset. After this, we use a pretrained model for the summarization task and compare the results with both the original and newly generated datasets. The results show that this method significantly improves the summarization of healthcare questions, producing summaries that are both fluent and informative, with potential benefits for better access to health information online.

Dataset

In this project, we aim to use the (MEQSUM) dataset . The MEQSUM (Medical Question Summarization) dataset is a specialized resource designed for evaluating question summarization

models in the healthcare domain. It was created specifically for tasks involving the generation of concise summaries from long or complex healthcare-related questions. The dataset focuses on generating summaries that preserve the key content while reducing redundancy and extraneous information. Its purpose is to support the training and evaluation of models capable of effectively summarizing medical and healthcare-related questions.

Dataset Structure

The MEQSUM dataset consists of a large set of healthcare-related questions, each paired with an expert-written, human-annotated summary. The dataset contains questions covering a wide range of medical topics, including but not limited to:

- **Symptoms:** Questions about symptoms of various diseases and conditions.
- **Treatments:** Questions about treatments for medical conditions.
- **Diagnosis:** Questions related to disease diagnosis processes and methods.
- **Medications:** Questions about medications, dosages, and side effects.

Each question in the dataset is designed to be a natural, real-world inquiry about health-related topics, with the associated summary condensing the original question into a simpler, more focused version that is easier to process.

Step 1: Load and Inspect the Dataset

- Load the dataset into a suitable data structure (such as a pandas DataFrame).
- Examine the dataset to understand its structure (e.g., columns like question, summary, etc.).

Step 2: Preprocessing

- Strip the text of any unnecessary characters such as special symbols, non-alphanumeric characters, extra spaces, or HTML tags that may be present.
- Convert all text (questions and summaries) to lowercase for consistency.

Step 3: Handling Missing or Irregular Data

- Identify and handle any missing or null values in the dataset (either by removing rows with missing data or filling them in).
- If questions or summaries are too long or too short (e.g., outliers), consider truncating or padding them to a specific length suitable for the model.

Round-Trip Translation (RTT)

Round-Trip Translation (RTT) is a method that involves translating text from one language to another and then back to the original language. This process is used to generate diverse paraphrases of the original text, which helps increase the variety of the dataset for training, especially in low-resource settings where large annotated datasets are unavailable.

Task Overview

The goal of applying RTT is to augment the dataset with diverse paraphrased versions of each healthcare-related question. By using a pivot language (i.e., a language other than English), questions are translated to that language and then back to English. This results in variations of the original question, which can improve the model's ability to generalize and summarize effectively.

Step 4: Translate Questions to a Pivot Language

- You should translate the questions to 5 different languages. Spanish (es), German (de), Italian (it), Chinese Simplified (zh), and French (fr).
- Use a machine translation model (e.g., MarianMTModel) to translate each selected healthcare-related question from English to the chosen pivot language.
- Use an API to translate the questions using Google Translate. (Bonus)

Step 5: Translate Back to English

- After obtaining the translated question in the pivot language, use the same machine translation model to translate the question back from the pivot language to English. This step will result in a new version of the original question that may have slight differences in phrasing or structure due to the round-trip translation process.

Question Selection

The next step is to select diverse and informative questions after using RTT for data augmentation. This will be done using three different methods:

- Frechét Question Distance (FQD)
- Precision Recall Question Distance (PRQD)
- Question Semantic Volume (QSV) - Bonus

Frechét Question Distance (FQD)

FQD measures the distributional distance between the semantic representation of the gold question (original question) and the round-trip generated question. We assume that question semantic representations follow the multidimensional Gaussian distribution with first two moments: mean and covariance. The distance between these two Gaussian distributions is measured by the Fréchet distance.

Let $\mathbf{h}_Q$ be the semantic representation of the gold question and $\hat{h}_Q$ be the semantic representation of the round-trip question.

The Fréchet Question Distance between Q and $\hat{Q}$ is computed as follows:

$$d_{\text{FQD}}(Q, \hat{Q}) = 1 - \frac{\mathbf{h}_Q \cdot \hat{h}_Q}{\|\mathbf{h}_Q\| \|\hat{h}_Q\|} \tag{1}$$

To produce a uniform FQD score, we linearly scale the $d_{\text{FQD}}(Q, \hat{Q})$ in the range [0, 1] using the following min-max normalization:

$$\text{FQD}(Q, \hat{Q}) = \frac{d_{\text{FQD}}(Q, \hat{Q}) - \min(d_{\text{FQD}})}{\max(d_{\text{FQD}}) - \min(d_{\text{FQD}})},$$

where $\min(d_{\text{FQD}})$ and $\max(d_{\text{FQD}})$ represent the minimum and maximum FQD in the dataset. When the distribution of the gold question is close to the distribution of the round-trip generated question, the FQD score is close to zero. In order to have the diverse, informative, and non-redundant samples in the training set, one does not need to include the round-trip generated questions whose FQD scores with gold questions are either low (near same question) or high (entirely different questions). Toward this, we aim to select the round-trip generated questions such that the FQD score with gold questions is found to be in an optimal range. Given the round-trip generated questions $D^{\text{rtt}}_{\text{en}\leftrightarrow\text{xx}}$ with pivot language (xx), we select a subset of the questions as follows:

$$D^{\text{en}\leftrightarrow\text{xx}}_{\text{rtt}+fqd} = \{(\hat{Q}_i, S_i) \mid \mu_1 < \text{FQD}(Q_i, \hat{Q}_i) < \mu_2\},$$

where μ_1 and μ_2 are hyper-parameters (i.e., the optimal threshold).

Step 6: Use FQD to select a subset of the new dataset

- For each healthcare-related question in the dataset, both the original question and the round-trip translated versions are passed through a pre-trained model (such as BERT) to obtain the semantic embeddings of the questions.
- compute the Frechét Distance between the distributions of the embeddings of the original question and the round-trip translated question.
- After calculating the Frechét distances for all pairs of questions (original and translated), you can sort the questions by their diversity and select a subset of questions using the provided formula.

Precision Recall Question Distance (PRQD)

Similar to the *FQD*, it measures the distributional distance between semantic representations of the gold and round-trip generated questions; however, it does not require estimating the moments of the probability distributions.

Intuitively, precision measures how much of $\hat{h}_Q$ can be generated by a portion of h_Q. In contrast, recall measures how much of h_Q can be generated by a portion of $\hat{h}_Q$. Hence, the precision and recall should be high for approximately the same question distributions, whereas, if the question distributions are disjoint in nature, the precision and recall will be zero. Therefore, we aim to select the RTT questions whose precision and recall lie between the optimal range to ensure diversity.

To compute PRQD, we follow the algorithm below, which is based on the precision-recall distance (PRD) curve. Toward this, we compute pairs of precision $\text{prec}(\alpha)$ and recall $\text{rec}(\alpha)$ for an equiangular grid of values of α:

$$\text{prec}(\alpha) = \sum_{v \in V} \min(\alpha h_Q(v), h_{\hat{Q}}(v)) \tag{2}$$

$$\text{rec}(\alpha) = \sum_{v \in V} \min\left(h_Q(v), \frac{\hat{h}_Q(v)}{\alpha}\right) \tag{3}$$

To compute a single-value metric, we calculate the F1-score corresponding to each α and select the maximum F1-score as the PRQD distance $d_{\text{PRQD}}(\hat{Q}, Q)$ as follows:

$$d_{\text{PRQD}}(\hat{Q}, Q) = \max_{\alpha \in \Lambda} \left(\frac{2 \cdot \text{prec}(\alpha) \cdot \text{rec}(\alpha)}{\text{prec}(\alpha) + \text{rec}(\alpha)}\right) \tag{4}$$

Step 7: Use PRQD to select a subset of the new dataset

- Embed the candidate and reference questions using an embedding model (like BERT). The embeddings should be vector representations of each question.
- Calculate precision based on reference questions and the candidate question (according to cosine similarity, Euclidean distance, or other similarity measures.)
- Calculate recall by measuring how many of the candidate question's relevant aspects (i.e., the nearest reference questions) are retrieved from the reference set.
- Combine precision and recall into a single PRQD score, using the provided formula.
- Sort the questions by their scores and select the best subset of questions.

Question Semantic Volume (QSV) - Bonus

Studies show that sentences that maximize the semantic volume in a distributed semantic space are the most diverse and have the least redundant sentences. For the given gold question Q and a set of K RTT generated questions $\{\hat{Q}_1, \hat{Q}_2, \ldots, \hat{Q}_K\}$, first, we extract the semantic representation $\mathbf{h}_Q$ for the gold question and each RTT question $\{\mathbf{h}_{\hat{Q}_1}, \mathbf{h}_{\hat{Q}_2}, \ldots, \mathbf{h}_{\hat{Q}_K}\}$ and form a data matrix $H \in R^{(K+1)\times d}$. Later, we perform the linear dimensionality reduction using Principal Component Analysis (PCA) to project the data matrix H to a lower-dimensional space and obtain the transformed data matrix $H' \in R^{(K+1)\times 2}$. In order to find and compare the most diverse round-trip candidate questions, we exclude the point corresponding to the gold question from H.

To find a convex maximum volume, we find the convex hull using the Quickhull algorithm as follows:

$$\{p_1, p_2, \ldots, p_C\} = \text{ConvexHull}(h'_1, h'_2, \ldots, h'_K)$$

The convex hull is the smallest convex set that includes all points $h'_1, h'_2, \ldots, h'_K$. The points $\{p_1, p_2, \ldots, p_C\}$ are the vertices of the convex hull. It also guarantees to obtain the maximum semantic area with the selected points. Intuitively, it selects the RTT questions which are diverse in nature.

Step 8 (Bonus): Use QSV to select a subset of the new dataset

- Convert each question into a vector representation using a model like BERT or another language model capable of generating dense, contextual embeddings.
- Calculate the semantic volume by measuring how well the questions spread out in the embedding space. Compute the covariance matrix of the question embeddings, which represents how the questions are distributed along different dimensions in the space.
- The semantic volume can be approximated by calculating the determinant of this covariance matrix. The larger the determinant, the larger the volume of the semantic space spanned by the set of questions. This value can serve as the semantic volume of the set of questions.
- Normalize the result based on the size of the embedding space to compare across different models or embedding sizes.
- Sort the questions by their semantic volume and select the best subset of questions.

Summarization

Step 9: Use pre-trained models to summarize questions

- Select a model, such as ProphetNet, BART, T5, or GPT Models for the summarization task.
- Use the model to generate summaries for the new datasets that you created in each of the previous steps.
- Use the model for the raw dataset.
- Try different models and compare their performances to determine which one yields the best results. (Bonus)

Evaluation

Step 10: Use evaluation metrics and compare the results

- Evaluate the performance of the generated summaries using standard evaluation metrics like Rouge-1 and Rouge-2.
- Compare the results.
- Use other metrics such as BLEU or METEOR. (Bonus)

Notes

Please carefully follow the instructions below to ensure you meet all requirements for this assignment. Failing to adhere to these instructions may result in a grade reduction.

1. Each step of the assignment has its own score. Ensure you complete every step thoroughly. Skipping any step will lead to a deduction in your grade.
2. Any form of cheating or copying in your work will result in a complete loss of grade.
3. Your code must include appropriate comments explaining the functionality of each section. Avoid generic or AI-generated comments. Comments should provide meaningful insights into your code.
4. **Complete Notebook with Outputs:**
    - You must provide a complete Jupyter notebook that includes the output of your code for **each section**.
    - Ensure all outputs are clearly visible. The absence of any required output will result in a grade deduction.
    - Present the notebook in a professional and visually appealing format.
5. **Detailed Report:**
    - Provide a comprehensive report explaining your code, the methods used, and the reasoning behind your choices.
    - Do not assume any step is obvious. Each detail must be included in the report to help me evaluate your work.

- The report is heavily weighted in the grading. An excellent report may earn you bonus points.

6. **Questions and Clarifications:** If you have any questions or need clarification at any point, do not hesitate to contact me (**Narges**). Do not make assumptions that could negatively impact your grade.

7. **Submission Requirements:**

   - Submit your work in the following format:
     (a) A **ZIP file** containing:
         – The Jupyter notebook.
         – The detailed report.
     (b) A separate **Python file** (.py) containing all of your code.
   - Organize the files in a clear and logical structure.

8. This project may differ slightly from what you learned in the course, but you can certainly approach it using the fundamentals you've learned in class, along with the detailed explanations I have provided. This is a real-world problem, highly relevant in the NLP field, and the skills you develop here can be valuable additions to your CV in the future. So, take this project seriously and aim to produce something meaningful, not just a simple solution to pass.

Good luck!