Group Project Report
AI News Reader

Group-05
Arjun Bingly
HaeLee Kim
Nayaeun Kwon

DATS-6312
Natural Language Processing
Dr. Amir Jafari

December 11, 2023

Abstract

Our Smart News Reader employs advanced Natural Language Processing (NLP) to transform news consumption. With features like news parsing, summarization, zero-shot classification, keyword extraction, and conversational question-answering, it streamlines information retrieval. Despite encountering challenges in implementing the translation component, the application represents a leap in redefining personalized and dynamic news exploration. Through continuous integration of NLP techniques, it addresses information overload, catering to evolving user needs in the digital age.

Contents

1 Introduction

1.1 Background

The escalating volume of digital content in today's information age necessitates the development of advanced tools for efficient information consumption. As the digital landscape expands, users face the challenge of navigating vast amounts of textual information. In response to this, the Smart News Reader application emerges as a solution to streamline news exploration, leveraging cutting-edge Natural Language Processing (NLP) techniques. This application aims to enhance the user experience by providing features such as summarization, question-answering, language translation, zero-shot classification, and keyword extraction. Through the integration of these features, users can gain deeper insights into news articles, breaking down language barriers, condensing lengthy content, and extracting key information. The application's foundation lies in addressing the growing demand for more effective and interactive methods of consuming digital news.

1.2 Objectives

- Introduce the Smart News Reader application.
  The Smart News Reader application is designed to revolutionize the way users interact with news articles, offering a user-friendly interface and a suite of NLP features.

- Highlight the importance of NLP in news exploration.
  In the era of information overload, NLP plays a crucial role in enhancing the efficiency and effectiveness of news exploration. The Smart News Reader leverages NLP techniques to empower users with tools that facilitate content summarization, language translation, dynamic question-answering, and more.

2 Features

2.1 News Parsing

For obtaining the news article body given a URL, the first attempt was to use BeautifulSoup to parse the HTML and extract the text body, but this proved to be difficult since each news website uses different structuring. This would restrict our app to a handful of websites. Upon looking further into this, we used the Newspaper3k python package which parses the News article given the URL. This extensive python package supports a lot of news websites and has rich features which includes extracting information like images, etc.

2.2 Summarization

The Summarization component plays a pivotal role in distilling extensive news articles into succinct and informative summaries. Leveraging advanced natural language processing techniques, the Summarizer class utilizes the BART-Large-CNN model. This model, pre-trained on the CNN news Dataset, since this was the most extensive news article dataset avaialble, we did not have to fine-tune the model.

However, the limitation of this model is the maximum token size. We observed that most news article are larger than the maximum allowed text input size but the summarization was still really good. To make sure that our summarizer does not leave out any critical information from the news article, LangChain was used to split the text into chunks of max token size, with a small overlap (in the hope to not loose the context). These chunks were then passed to the model and the summaries combined. This seems to work really well, however the downside is that, we no longer have direct control over the max summary size.
One was to solve this problem would be recursively summarize the long summaries, but we encountered a longer run-time for this implementation. Moreover, a single error in the first summarization would be exaggerated in the recursive summarization.

2.3 Zero-Shot Classification

The main idea behind this module is that the users will be able to check if a particular news article is about the a particular personalized label category.

The Zero-Shot Classification module addresses the challenge of categorizing news articles without the need for specific training data for each category. Leveraging the BART-Large-MNLI model fine-tuned on CNN (AyoubChLin/Bart-MNLI-CNN_news), integrated into the ZeroShotClassifier class, this module excels in assigning relevant labels to news articles.

2.4 Keyword Extraction

The idea behind key-Word extraction is to give a much smaller overview to the users about the main idea about the article.
Although there are multiple methods for doing this including tf-idf, YAKE, etc. ,these models are statistical and does not take context into consideration. An alternative that takes context, is using a transformer based model for key-phrase generation but this model does not guarantee that the key-phrase exists in the main body.
So we used the python package called KeyBERT, this uses BART sub-word embeddings and with simple cosine similarity finds the word/phrase that is most similar to the document.

2.5 Conversational QA

The idea behind this feature is to enable the users to ask questions regarding the article and get concise answers to it. But we wanted this to be conversational, instead of comprehension answering.

Initially, we implemented the Bert-Large fine-tuned on SQUAD dataset. There were many problems with this implementation including, truncation due to max token limits and the model being non-conversational. The truncation was tacked using Retrieval Augmented Modeling using LangChain, this is implemented by first splitting the news article into smaller chunks with overlap and then storing their embedding in a vector database (Chroma in our case). When the question is input, the most similar chunks are provided to the model as context to answer the question. This works well but we found that the embeddings used to store the vector plays a crucial part in the performance of the model. In our trial, we found 'jinaai/jina-embeddings-v2-base-en' to perform the best but this could not be incorporated into LangChain as HuggingFace requires a user input 'yes' to run external code and the LangChain wrapper does not have a way to incorporate this explicitly. So we settled on the next best model we found, 'sentence-transformers/all-MiniLM-L12-v2'. However, this model is still not conversational and the answers provided were mostly just phrases or at most sentences from the text.

To have a conversational model we had to turn to larger LLM models and since we wanted to run everything locally, we did not use any API based approach, instead we got Llama-2 model by META after submitting a request form online. We initially tried using Llama-2 7B chat, this model was small enough to fit on the A10 GPU but did not give impressive results, so we tried the 13B chat model, but this model was too large for the GPU. Hence, we tried quantizing the model to 4-bit integer precision using llama.cpp but we encountered dependency issues while compiling the git-repo. Therefore, we turned to already quantized models on github and followed the recommendations in the LangChain documentation and used TheBloke/Llama-2-13B-chat-GGUF on HuggingFace. The implementation of this model is quite different to the QnA model since this is purely run using LangChain and llama-cpp-python. Llama-cpp is just an interface to run the model, we used CuBLAS to enable inferencing on the GPU. LangChain's Conversational Retrieval Chain is used, this chain utilizes the following prompt to condense follow up questions using chat history to a stand-alone question and then,

```
Given the following conversation and a follow up question , rephrase the
    follow up question to be a standalone question , in its original
    language .
```

```

Chat History: {chat_history}
Follow Up Input: {question}
Standalone question:"""
```

the following prompt is then used to generate the answer:

```
Answer the question based only on the following context:{context}

Question: {question}
```

Moreover, the memory feature from LangChain Memory Buffer is implemented to keep track of previous chat inputs and outputs. Also this model is capable of showing the user the specific parts from the article used to answer the question.

2.6 Translation - Failed to Implement

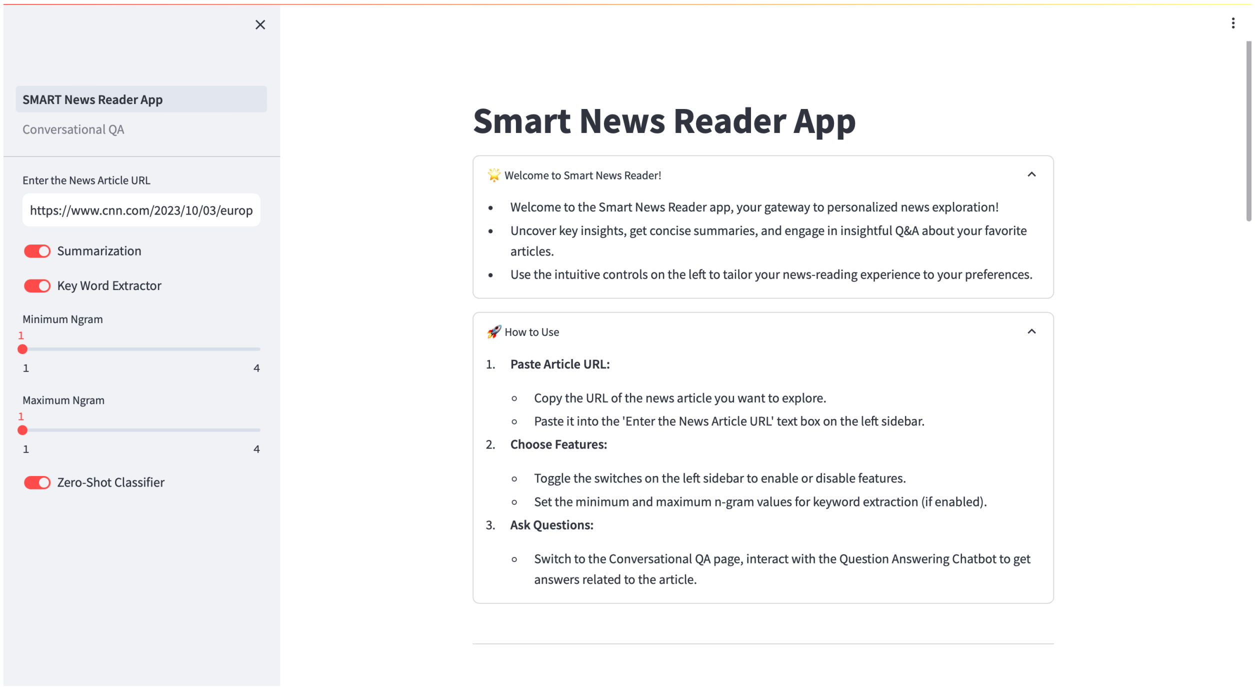
The Translation component ensures the accessibility of news articles across diverse linguistic audiences. The Many-to-One translation module, equipped with the MBART model, is proficient in translating news articles from various source languages to English. The NewsTranslator class dynamically handles language codes, offering an efficient and flexible translation mechanism.
Though the model works pretty well and the news fetch module gives us the source language, the implementation caused some dependency related issues and we could not implement it in time.

3 The App

Since all the coding was done in object-oriented method, implementation of the app using StreamLit relatively easy. The challenges included:

- **GPU Memory**: due to the limited GPU memory we had to resort to only running the summarizer and chat-bot on the GPU leaving everything else on the CPU. This is clearly not a production ready set-up due to high cost of GPU machines.
- **Whole App Refresh**: Due to the way StreamLit works the entire app re-runs top to bottom upon any change in the inputs, this is very costly for our app. Hence, a multi-page approach was taken since in a multi-page set up only the active page re-runs. The chat-bot, which is the most computationally expensive module in our app was moved to a separate page
- **Whole App Refresh**: Due to the way StreamLit works the entire app re-runs top to bottom upon any change in the inputs, this is very costly for our app. Hence, a multi-page approach was taken since in a multi-page set up only the active page re-runs. The chat-bot, which is the most computationally expensive module in our app was moved to a separate page
- **Caching**: Since the app reloads every time, re-loading all the models is very wasteful, hence the caching functionality offered by StreamLit was utilized.
- **Session States**: Session states are the only elements that can be carried over across re-runs, hence they were effectively used in the effort to preserve user inputs and also to pass user inputs and other elements across pages.

Figure 1 shows the about and help section of the main app page while *Figure 2* shows parsed news article with the news title, authors, original url, publishing date and also an image from the original article. *Figure 3* shows the generated summary, the original news article had about 1200 words while the generated summary has about 150 words; the figure also shows extracted key-words which are clearly aligned to the article. *Figure 4* shows the zero-shot classification and it rightly classifies the news article. *Figure 5* shows the conversational chat-bot page, you can see that the questions were well answered and the relevant parts from the article used by the chat model displayed.



Enter the News Article URL

https://www.cnn.com/2023/10/03/europ

● Summarization

● Key Word Extractor

Minimum Ngram: 1 (1 – 4)

Maximum Ngram: 1 (1 – 4)

● Zero-Shot Classifier

Smart News Reader App

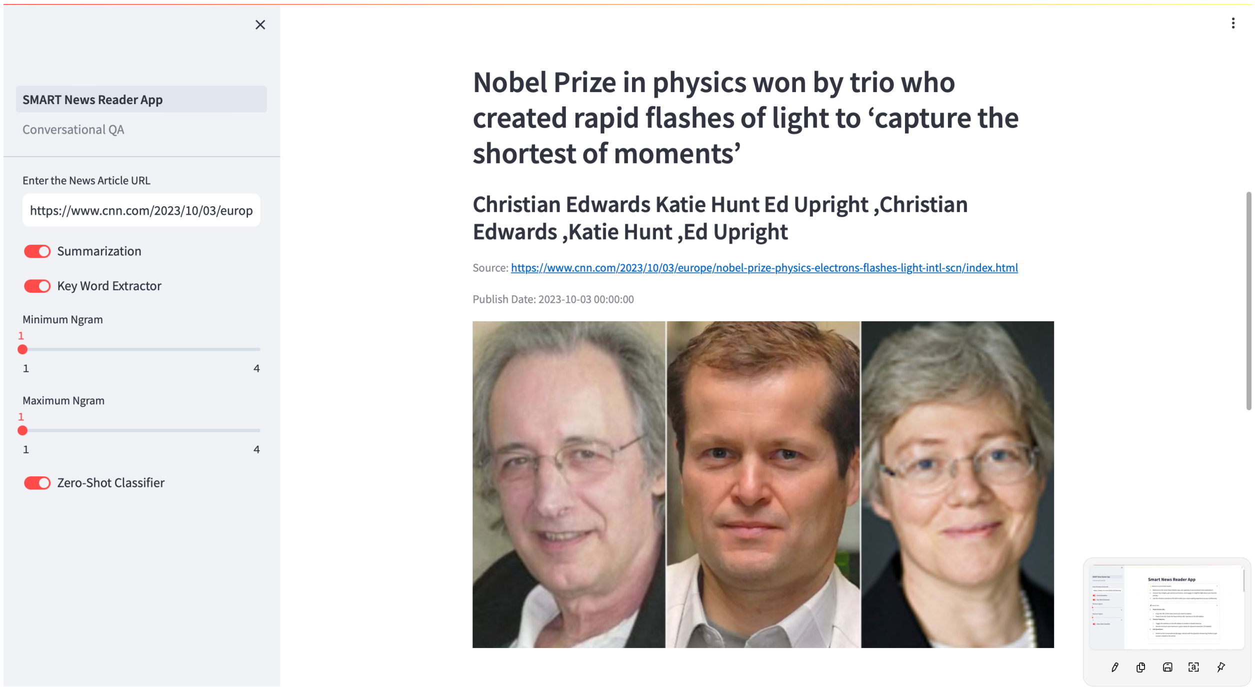
🌟 Welcome to Smart News Reader!

- Welcome to the Smart News Reader app, your gateway to personalized news exploration!
- Uncover key insights, get concise summaries, and engage in insightful Q&A about your favorite articles.
- Use the intuitive controls on the left to tailor your news-reading experience to your preferences.

🚀 How to Use

1. **Paste Article URL:**
   - Copy the URL of the news article you want to explore.
   - Paste it into the 'Enter the News Article URL' text box on the left sidebar.
2. **Choose Features:**
   - Toggle the switches on the left sidebar to enable or disable features.
   - Set the minimum and maximum n-gram values for keyword extraction (if enabled).
3. **Ask Questions:**
   - Switch to the Conversational QA page, interact with the Question Answering Chatbot to get answers related to the article.

Figure 1: Prediction Plot using the Regression Model



Enter the News Article URL

https://www.cnn.com/2023/10/03/europ

● Summarization

● Key Word Extractor

Minimum Ngram: 1 (1 – 4)

Maximum Ngram: 1 (1 – 4)

● Zero-Shot Classifier

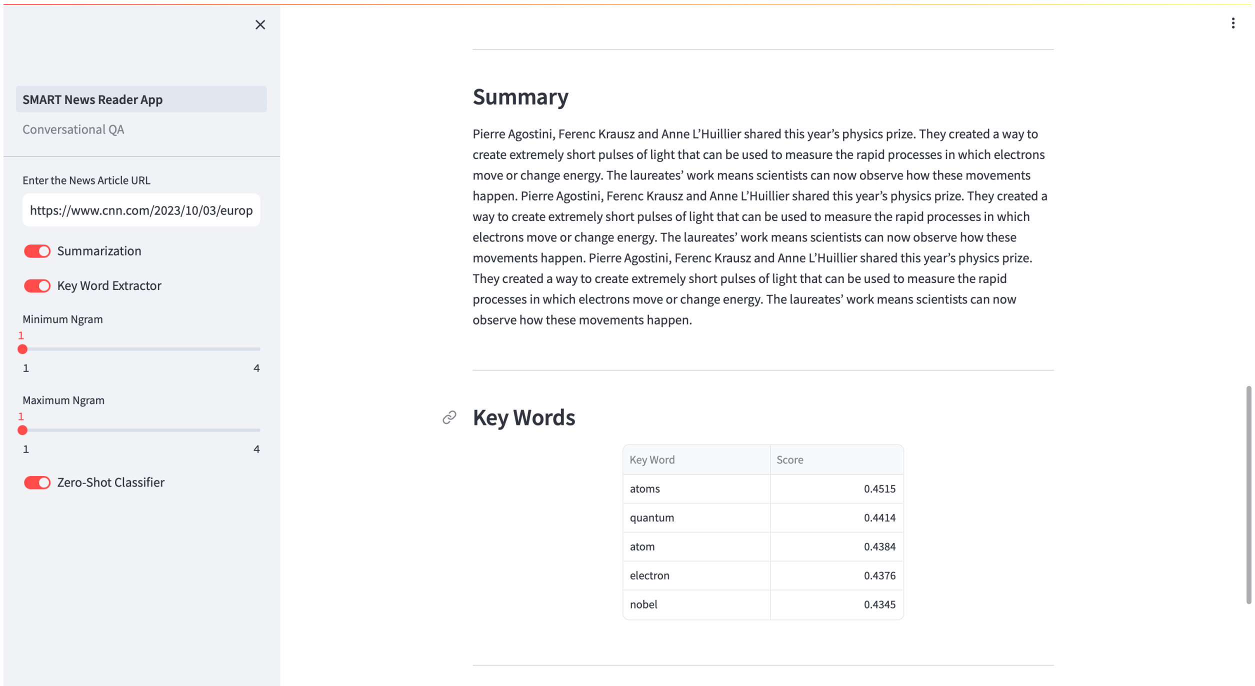
Nobel Prize in physics won by trio who created rapid flashes of light to ‘capture the shortest of moments’

Christian Edwards Katie Hunt Ed Upright ,Christian Edwards ,Katie Hunt ,Ed Upright

Source: https://www.cnn.com/2023/10/03/europe/nobel-prize-physics-electrons-flashes-light-intl-scn/index.html

Publish Date: 2023-10-03 00:00:00

Figure 2: Prediction Plot using the Regression Model



Enter the News Article URL: https://www.cnn.com/2023/10/03/europ

● Summarization

● Key Word Extractor

Minimum Ngram: 1 (1–4)

Maximum Ngram: 1 (1–4)

● Zero-Shot Classifier

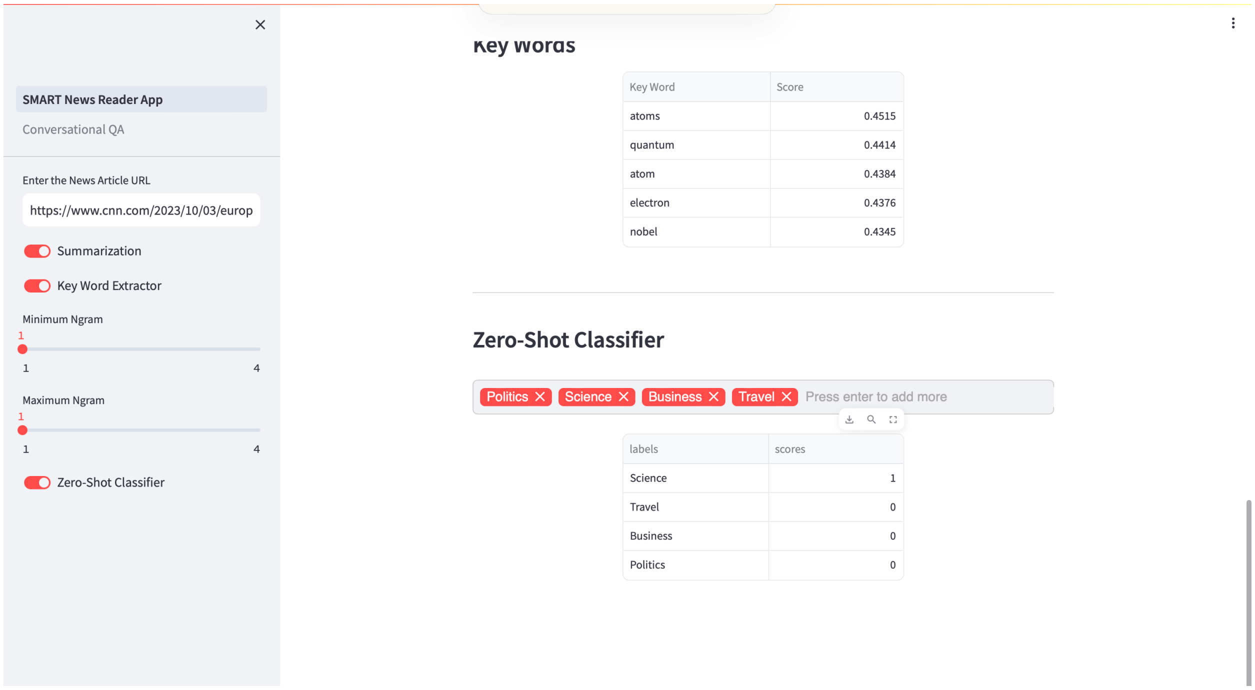
Summary

Pierre Agostini, Ferenc Krausz and Anne L'Huillier shared this year's physics prize. They created a way to create extremely short pulses of light that can be used to measure the rapid processes in which electrons move or change energy. The laureates' work means scientists can now observe how these movements happen. Pierre Agostini, Ferenc Krausz and Anne L'Huillier shared this year's physics prize. They created a way to create extremely short pulses of light that can be used to measure the rapid processes in which electrons move or change energy. The laureates' work means scientists can now observe how these movements happen. Pierre Agostini, Ferenc Krausz and Anne L'Huillier shared this year's physics prize. They created a way to create extremely short pulses of light that can be used to measure the rapid processes in which electrons move or change energy. The laureates' work means scientists can now observe how these movements happen.

Key Words

Key Word	Score
atoms	0.4515
quantum	0.4414
atom	0.4384
electron	0.4376
nobel	0.4345

Figure 3: Prediction Plot using the Regression Model



Enter the News Article URL: https://www.cnn.com/2023/10/03/europ

● Summarization

● Key Word Extractor

Minimum Ngram: 1 (1–4)

Maximum Ngram: 1 (1–4)

● Zero-Shot Classifier

Key Words

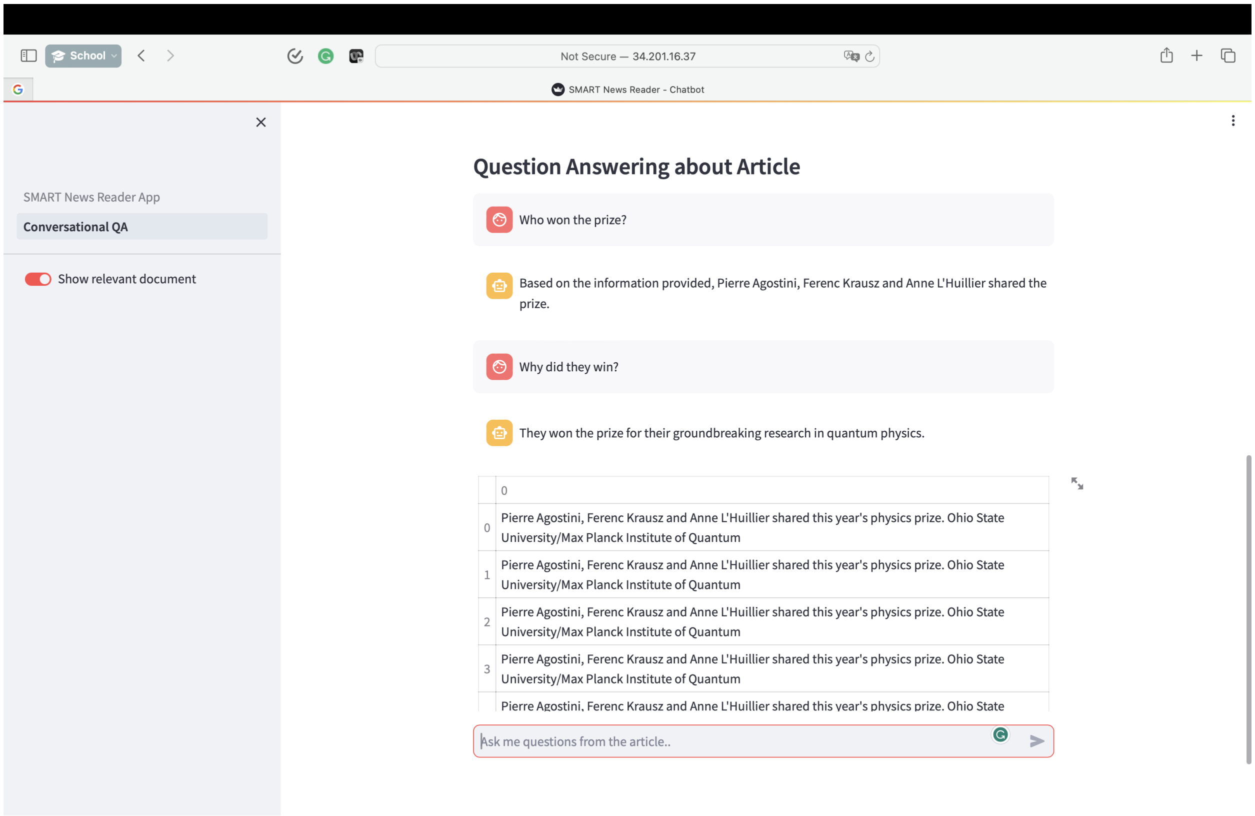
Key Word	Score
atoms	0.4515
quantum	0.4414
atom	0.4384
electron	0.4376
nobel	0.4345

Zero-Shot Classifier

Politics ✕ Science ✕ Business ✕ Travel ✕ Press enter to add more

labels	scores
Science	1
Travel	0
Business	0
Politics	0

Figure 4: Prediction Plot using the Regression Model


School
Not Secure — 34.201.16.37
SMART News Reader - Chatbot
SMART News Reader App
Conversational QA
Show relevant document
Ask me questions from the article..

Question Answering about Article

Who won the prize?

Based on the information provided, Pierre Agostini, Ferenc Krausz and Anne L'Huillier shared the prize.

Why did they win?

They won the prize for their groundbreaking research in quantum physics.

	0
0	Pierre Agostini, Ferenc Krausz and Anne L'Huillier shared this year's physics prize. Ohio State University/Max Planck Institute of Quantum
1	Pierre Agostini, Ferenc Krausz and Anne L'Huillier shared this year's physics prize. Ohio State University/Max Planck Institute of Quantum
2	Pierre Agostini, Ferenc Krausz and Anne L'Huillier shared this year's physics prize. Ohio State University/Max Planck Institute of Quantum
3	Pierre Agostini, Ferenc Krausz and Anne L'Huillier shared this year's physics prize. Ohio State University/Max Planck Institute of Quantum
	Pierre Agostini, Ferenc Krausz and Anne L'Huillier shared this year's physics prize. Ohio State

Figure 5: Prediction Plot using the Regression Model

4 Conclusion

The Smart News Reader application represents a significant leap forward in redefining the landscape of news consumption. By addressing the challenges posed by the increasing volume of digital content, the application introduces a user-centric approach to news exploration. The incorporation of advanced NLP features, including summarization, question-answering, language translation, zero-shot classification, and keyword extraction, empowers users with versatile tools for extracting, comprehending, and interacting with news articles.

As we have explored in-depth, the Summarization component condenses lengthy articles into informative summaries, the Zero-Shot Classification module categorizes articles dynamically. Furthermore, the Keyword Extraction and Question-Answering features contribute to a more insightful and interactive news exploration experience.

In conclusion, the Smart News Reader not only addresses the immediate challenges of information overload but also anticipates the evolving needs of users in an increasingly digital world. The continuous integration of state-of-the-art NLP techniques ensures that the application remains at the forefront of facilitating efficient and engaging news consumption. Through this innovative approach, the Smart News Reader sets a new standard for personalized and dynamic news exploration.

5 Possible Improvements

- The whole app takes a lot of computational resource and takes a fairly long time to run, optimizations of both computational resource and time could be done.
- The app uses multiple models, this could be condensed to a single LLM.
- Every input refreshes the whole app, even with caching more improvements needs to be made.
- Better UI.
- Implementation of the translation feature.
- App currently only supports one user.