FINAL REPORT:

HHM'S QUESTION ANSWERING WEB EXTENSION

TEAM: 24 | Date 30/05/2021

[ISYS2101] SOFTWARE ENGINEERING PROJECT MANAGEMENT

Supervisor: Anna Filipe

Members:

1. Dang Ba Minh - s3685119
2. Le Nguyen Minh Huy - s3777280
3. Nguyen Quang Huy – s3697272

I declare that in submitting all work for this assessment I have read, understood and agreed to the content and expectations of the assessment declaration.

Table of Contents

Table of Figures

Table of Tables

I

Product description & functional block diagram

1. Product description

As the amount of online content continues to explode, people need a better way to be more productive and save more time while obtaining online information. Inspiring by the reading and understanding natural language ability of the machine, the team decides to develop a Question Answering Web extension named HHM's QA. With our application, the users can get information from a particular website faster. Specifically, when they are on a website and want to get information from it, they only need to open our application and insert their questions. The extension then parses the web content along with the questions. If the answer is available, it will return it. Thus, the users can retrieve the information they want without reading the entire content.

Throughout the 3-month development process, the team has covered all essential knowledge for this project and achieved the initial goal and the expected outcome. The extension has two main functionalities: asking questions and rating answers. For the asking feature, we train a deep learning model for the question answering task and develop a service to serve it. If the answers are available, the extension will return the best three; otherwise, it displays a message to announce that there is no answer. On the other hand, the rating feature helps the team collect more data from users. We can later use those data to improve the model performance.

For the communication between the extension and the server, we develop two Application Programming Interfaces (APIs). We use one API to send the question and context to the server and receive the answer as a response. We use another one to send the user feedback. Specifically, the extension sends the rated answer along with the question and web content to the database server.

2. Functional description

- **Inserting the question:**

Users are required to type their questions to the input in the extension. However, Since the model learns from the dataset with "Wh-questions," the extension will return a better result if the users ask the questions in the same format. Thus, we have an input placeholder to guides the users on how to make questions appropriately.

- **Answering the question:**

After receiving the questions from users and the corresponding context - the webpage content, the extension will send them to the prediction service to get the answers and display them. Also, the application automatically scrolls the web page to the first answer and highlights it.

- **Discovering the answers:**

As the system returns the most three correct answers, this feature enables users to scroll to the place containing the chosen answer by clicking on it.

- **Rating the most appropriate answer:**

The extension implements this feature to enhance the user experience in the future. Thanks to the rating data, the developer team can make some adjustments by trying different Machine Learning algorithms or implementing different architectures to improve the performance. To rate the answer, the users need to click on the "star" button next to each answer.

3. Software architecture

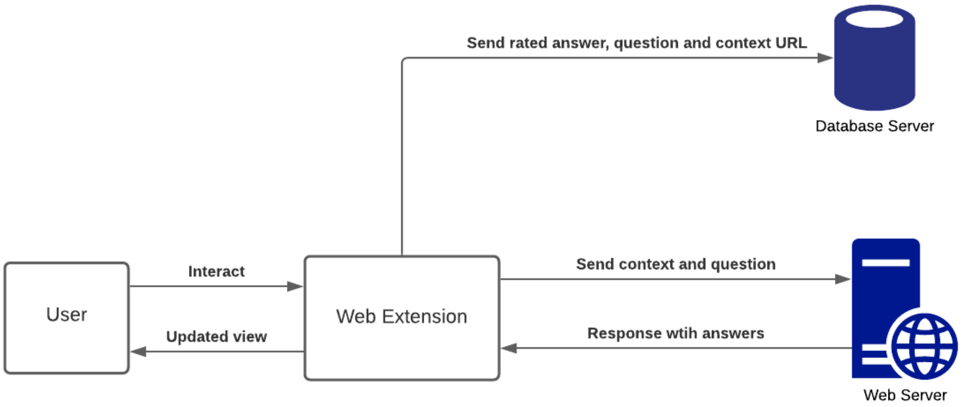


Figure 1: System Architecture

There are three main components in the software architecture of our system: web extension, web server, and database server. The user has three ways to interact with the system via the web extension.

Firstly, the user can insert his question(s) to the extension by typing the text input to the UI. After the user hits "Enter," the extension will process the web content (context) and send it along with the inserted question to the prediction service in the webserver. The prediction service then generates the answers based on the received input and sends it back to the extension. The extension uses them to update the UI.

Secondly, the user can choose to highlight a particular answer displayed in UI by clicking on it. The extension then scrolls the web page to the position of that answer and highlights it.

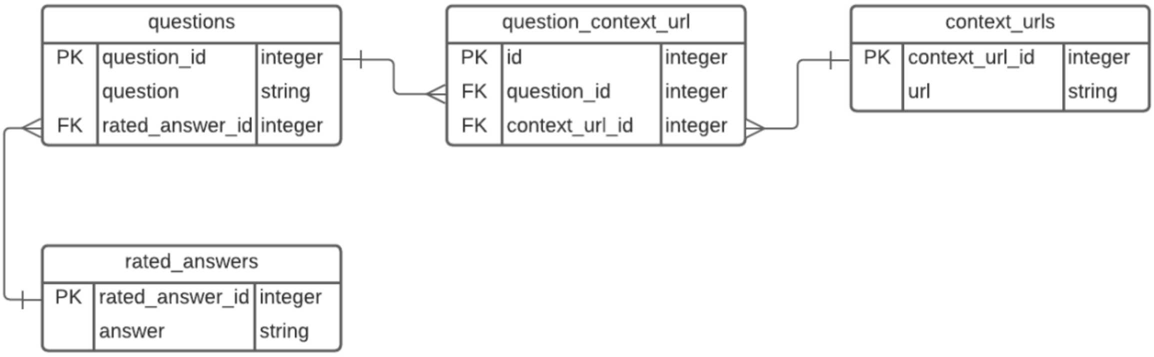
Finally, the user can rate the best answer for his question by clicking on the "star" beside the answer. The extension then sends the rated one together with the current question and the context URL to the Database server to store. This information serves as user feedback and will be used to improve the performance of the prediction service in the future.

II

Technical specification

1. Data model diagram

To store user feedback, we create a database called "QA database." The database has only three entities: questions rated answers and context URLs. Thus, we create three tables called “questions”, “rated_answers” and “context_urls” corresponding to three entities. Besides, since the relationship between questions and context URLs is many-to-many, we also create a *joint table* called “question_context_url”, which contains the ids of the two tables.



questions		
PK	question_id	integer
	question	string
FK	rated_answer_id	integer

question_context_url		
PK	id	integer
FK	question_id	integer
FK	context_url_id	integer

context_urls		
PK	context_url_id	integer
	url	string

rated_answers		
PK	rated_answer_id	integer
	answer	string

Figure 2: Dataflow diagram

2. Use case diagram

A use case indicates a unique functionality of a system accomplished by a user. Its purpose is to capture the functions of a system and demonstrate the interaction of actors with the use case [1]. The use case diagram below depicts the interaction between our Question Answering Web Extension with its users: a human, web server, and database server.

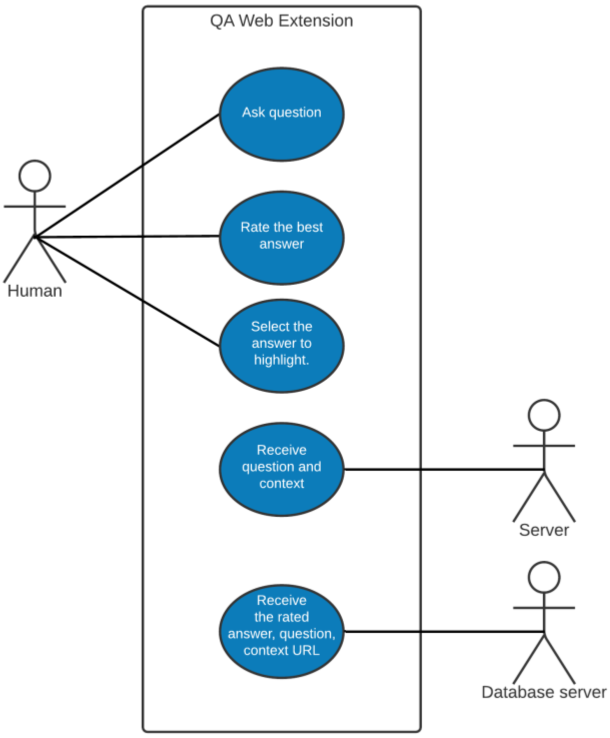

QA Web Extension
Ask question
Rate the best answer
Select the answer to highlight.
Receive question and context
Receive the rated answer, question, context URL
Human
Server
Database server

Figure 3: Use case diagram

The human user can ask questions, rate the best answer and highlight a particular one. The database server can receive the rated answer(s), question(s), and context URLs, while the web server can receive question and context from the extension.

3. Sequence diagram

Sequence diagram describes **how** and **in what order** a group of objects including human, prediction service, and QA database interact with each other [2].

3.1. Sequence diagram for asking question use case

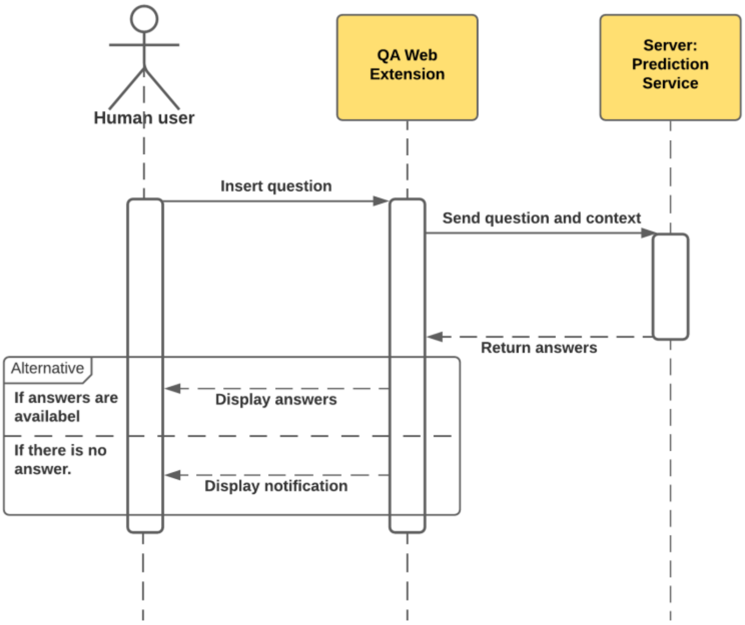


Figure 4: Sequence diagram for asking question

The human user (actor) inserts the question to the web extension. The extension then sends question and processed context to the prediction service in the web server. After processing the input, the service responses with the answers. If the answers are available, the UI will be updated with those answers, otherwise, it will display a message to notify the user that there is no answer for his question.

3.2. Sequence diagram for highlighting an answer use case

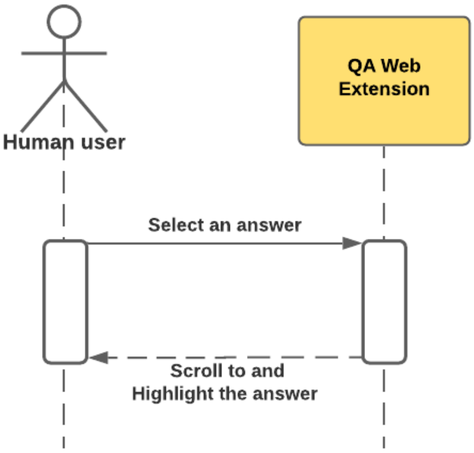


Figure 5: Sequence diagram for highlighting an answer

The human user (actor) inserts the question to the web extension. The extension then sends question and processed context to the prediction service in the webserver. After processing the input, the service responds with the answers. If the answers are available, the UI will update with those answers; Otherwise, it will display a message to notify the user that there is no answer to his question.

3.3. Sequence diagram for rating an answer use case

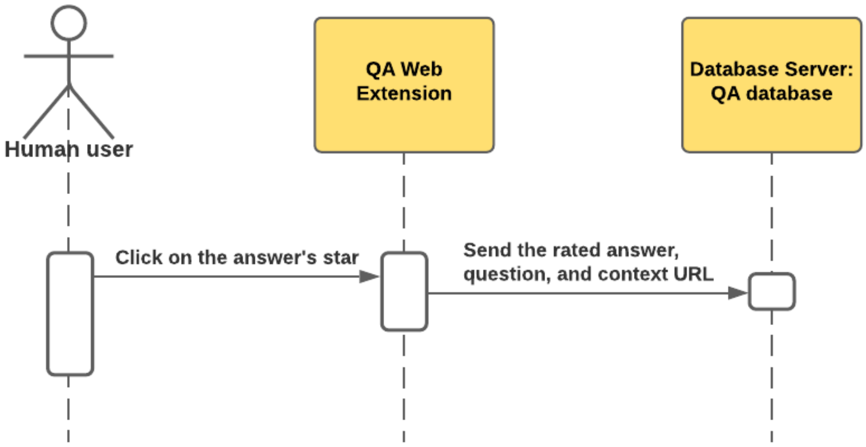

Human user
QA Web Extension
Database Server: QA database
Click on the answer's star
Send the rated answer, question, and context URL

Figure 6: Sequence diagram for rating an answer

After the human user (actor) clicks on the answer’s star, the extension will send that answer, current question, and context URLs to the QA database in the database server.

4. Activity diagram

The activity diagrams describe the steps performed in our use case diagram.

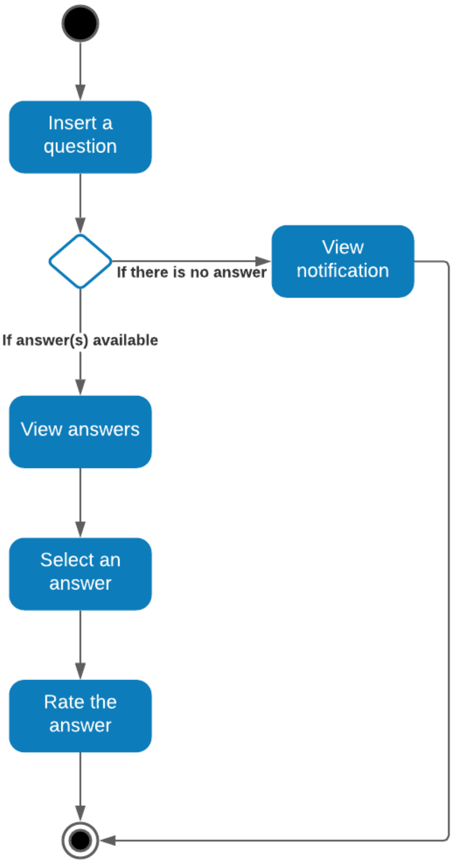


Figure 7: Activity diagram

After opening the extension, the user can type in a question he wants to ask for the current website. The extension then sends it along with the context to the prediction server to get the answers. If the answers are available, the extension, then display; otherwise, it will notify the user that there is no answer to his question. Thus, depending on the situation, the user can view the answers or the notification. In the former case, he then can select one of the answers, and the extension will scroll to and highlight it. After checking all of the answers, the user can rate the best one.

5. Question Answering system specification

5.1. Background

5.1.1 Question Answering System

In a typical reading comprehension or question answering (QA) system, we input a combination of a context and a question and receive the answer(s) for that question. The goal is to provide the correct answer.

In this project, we apply the question answering system to create the web extension used to extract information from web pages. Even though we have seen tremendous progress in a machine reading comprehension problem in recent years [3], it is still not adopted widely in the industry due to the generalization problem. Specifically, the model struggles to make correct predictions in the domains, which do not have a training dataset [3]. That makes the model hard to apply in the field where the training dataset is not available or difficult to collect. For our project, we tried two techniques, Task Adaptive Pretraining (TAPT) and Data Augmentation, to improve the model's generalizability.

5.1.2 Task Adaptive Pretraining

Task Adaptive Pretraining is a technique that helps the model adapt to a specific task by continuing to pretraining the model with data from downstream problems [4]. Specifically, we will jointly optimize the QA-based loss together with the Masked Language Model-based loss over the inputs. Many previous works have proven the effectiveness of this technique [5]. Thus, we decide to apply it to train our models.

5.1.3 Data Augmentation

Data Augmentation is a technique used to generate additional, synthetic data using the existing ones. It is a common and widely used technique for Computer Vision related tasks. In the past, it was hard to apply this technique to Natural Language Processing (NLP) tasks due to the complexity of languages. Changing a word in a sentence can completely change the meaning of the whole context. Yet, in the paper [6], they show a substantial gain in all tasks and achieve new State-of-the-art results via a back-translation technique.

Technically, a back-translation technique enables us to paraphrase each instance in the dataset. Specifically, we will translate the original corpus from English to different languages and then translate it back to English [7]. Furthermore, while doing experiments in the baseline model, the back translation technique is not only performed on each question, each context but also both question and context in the datasets. To implement the technique, the team has used the BackTranslation python package which utilizes *the googletrans* library and *Baidu Translation API*.

5.2. Approach

5.2.1 Dataset

We used three datasets including SQuAD, NewsQA, and Natural Questions as ***in-domain*** datasets, and three dataset including Relation Extraction, RACE, and DuoRC as ***out-domain*** datasets. We split each dataset into train and validation sets. We use the union of ***in-domain*** and ***out-domain*** train sets to train the models and evaluate them on ***in-domain*** and ***out-domain*** validation sets.

Table 1: Dataset summary

Dataset	Train	Dev
In-domain dataset		
SQuAD [8]	50000	10,507
Natural Questions [9]	50000	12,836
NewsQA [10]	50000	4,212
Out-domain dataset		
RACE [11]	127	128
RelationExtraction [12]	127	128

DuoRC [13]	127	126

All the datasets will be processed and transformed to the SQuAD format. According to the SQuAD format, each sample will have five features: answers, context, id, question, and title. The value of the answers feature is an object with two attributes: ‘answer_start’ and ‘text’. The value of the ‘answer_start’ attribute is a start index of the answer in the context, while the value of the ‘text’ attribute is the text answer.

Table 2: Data sample

answers	context	id	question	title
{'answer_start': [595], 'text': ['1964']}	Paul VI opened the third period on 14 September 1964, telling the Council Fathers that he viewed the text about the Church as the most important document to come out from the Council. As the Council discussed the role of bishops in the papacy, Paul VI issued an explanatory note confirming the primacy of the papacy, a step which was viewed by some as meddling in the affairs of the Council American bishops pushed for a speedy resolution on religious freedom, but Paul VI insisted this to be approved together with related texts such as ecumenism. The Pope concluded the session on 21 November 1964, with the formal pronouncement of Mary as Mother of the Church	5726bc075951b619008f7c63	In what year did Paul VI formally appoint Mary as mother of the Catholic church?	Pope_Paul_VI

5.2.2 Experiment details

Firstly, we finetune the DistilBERT model on the ***in-domain*** dataset to create the first baseline (**Baseline_01**). We later use this one to compare with the baseline trained by using the TAPT technique. Also, train the **Baseline_02** model by continuing training **Baseline_01** on the ***out-domain*** dataset. This baseline will be used to evaluate the effectiveness of the data augmentation technique.

To apply TAPT, we first train the DistilBERT with MLM task using the ***out-domain*** dataset. Specifically, we randomly mask out 15% of the token in each sentence. For the chosen token to be masked, 80% we replace it with [MASK] token, 10% we replace it with a random token, and the last 10% we keep it unchanged. This is the masking procedure from the original paper of the MLM task to reduce the effect of mismatch between pre-training and fine-tuning [Ref]. After finishing pretraining with the **out-domain** dataset, we fine-tune the model in the same way as **Baseline_01** to create the **TAPT_model**.

For data augmentation, we generate the datasets for the task using three ways. One is back-translating only the questions, backing-translate only the contexts, and back-translating both the questions and contexts. We use Google Translation API to do the translation. Thus, the performance of this technique heavily relies on that service.

In our experiment, because of the time constraint, we only apply the technique on the **out-domain** datasets. To be more specific, to augment the questions, we translate each of them into

7 common languages and then translate them back into English. For the context, things are more complicated due to the format of the datasets. Since the back translate technique will generate a new text based on the original one, it is more likely that the length of the new context will be different. Thus, the answer start index may no longer match the context. Also, during the back translation process, the answer text may also be replaced with its synonym. Therefore, instead of back translating the whole context, we only do it for sentences that do not contain the answer, which will help us keep the answer text unchanged in the context. Yet, we still have to modify to answer start index to match with the new context. After generating new datasets, we then apply them to both **Baseline_01** and **TAPT_model**.

Table 3: Experiment summary

Technique	Training Data	Epochs	Max Length	Eval-Every	Validating Data
	ind	3	384	20	in-val
	ood	10	384	20	ood-val
DA	ood + bt_question	10	384	20	ood-val
DA	ood + bt_context	10	384	20	ood-val
DA	ood + bt_question_context	10	384	20	ood-val
TAPT	ood	10	512		ood-val

5.2.3 Result summary

Table 4: Result summary

Model	Out of domain validation set	
	EM	F1
Baseline_01	33.25	48.43
Baseline_02	34.03	49.19
TAPT_model	33.77	49.56
BT_question_01	33.77	48.61
TAPT_BT_question	34.82	49.23
BT_context_01	34.3	49.87
TAPT_BT_context	35.08	50.68
BT_question_context_01	**35.34**	**50.82**
TAPT_BT_question_context	35.86	50.08

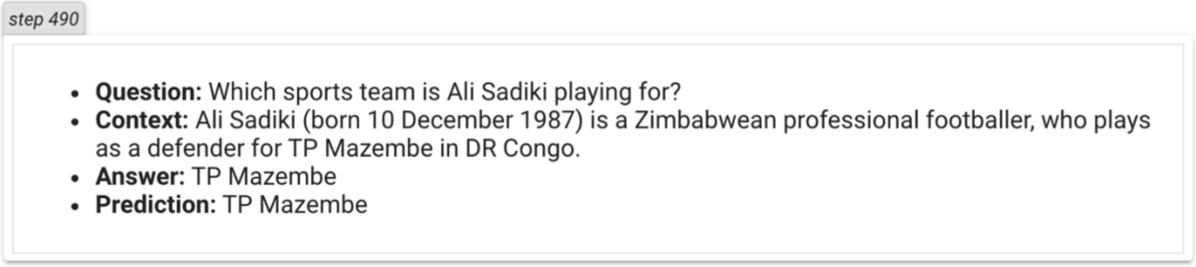
The table above has shown the main results of all approaches through the evaluation metrics for each model. The first baseline model (**Baseline_01**) achieves an F1 score of **48.43** and an EM score of **33.25** on the in-domain validation set. The second baseline model (**Baseline_02**) further training on **Baseline_01** has a slightly higher result on the out-domain dataset, in both F1 and EM score. Our implementation of the Task Adaptive Pre-training (TAPT) technique has only improved the EM score to **33.77** compared to the first baseline and achieves an F1 score of **49.56** on the validation set. On the other hand, with the augmented dataset, the **BT_question_context_01** model further training on the first baseline has achieved an EM score of **35.34** and F1 score of **50.82** on the validation set. This model is also the best one among our experiments.

5.3. Analysis

In this section, the team analyzes the model's performance in two scenarios: the success of cases and the failure of cases of the best model. Then, the team compares the performance of the final model with the baseline to demonstrates its improvement.

Success case analysis

Example 1:



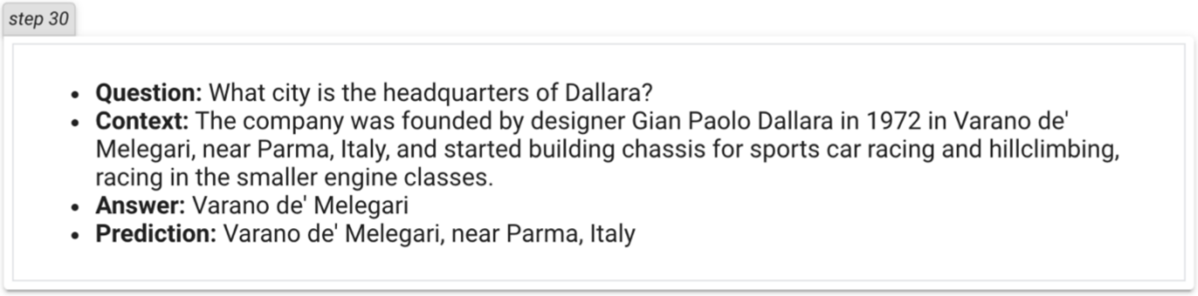
step 490

- **Question:** Which sports team is Ali Sadiki playing for?
- **Context:** Ali Sadiki (born 10 December 1987) is a Zimbabwean professional footballer, who plays as a defender for TP Mazembe in DR Congo.
- **Answer:** TP Mazembe
- **Prediction:** TP Mazembe

Figure 8: Success example 1

In this example, the model has made an accurate prediction, which interprets that it successfully learns the word "sports team" and the relation between "Ali Sadiki" and "TP Mazembe".

Example 2:



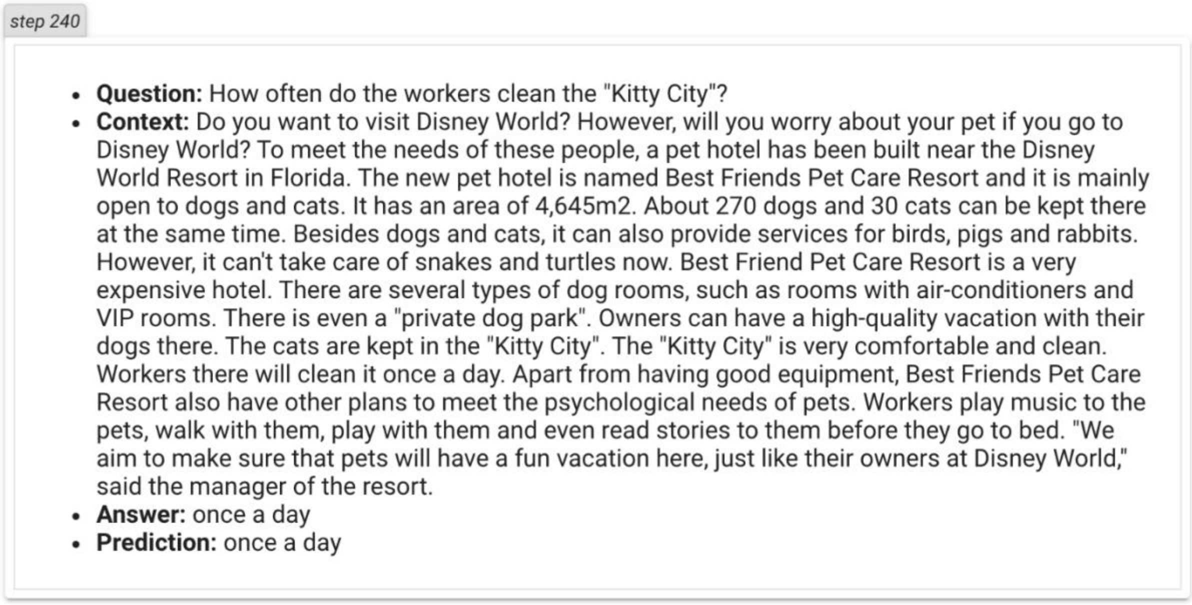
step 30

- **Question:** What city is the headquarters of Dallara?
- **Context:** The company was founded by designer Gian Paolo Dallara in 1972 in Varano de' Melegari, near Parma, Italy, and started building chassis for sports car racing and hillclimbing, racing in the smaller engine classes.
- **Answer:** Varano de' Melegari
- **Prediction:** Varano de' Melegari, near Parma, Italy

Figure 9: Success example 2

Regarding the second example, the model's prediction does contain the name of a city, "Varano de' Melegari" which is the right answer. However, it mistakenly also includes the geographical information of that city. This result indicates that the model is able to understand the question and recognize the name of a specific place but cannot abstain from redundant knowledge.

Example 3:



step 240

- **Question:** How often do the workers clean the "Kitty City"?
- **Context:** Do you want to visit Disney World? However, will you worry about your pet if you go to Disney World? To meet the needs of these people, a pet hotel has been built near the Disney World Resort in Florida. The new pet hotel is named Best Friends Pet Care Resort and it is mainly open to dogs and cats. It has an area of 4,645m2. About 270 dogs and 30 cats can be kept there at the same time. Besides dogs and cats, it can also provide services for birds, pigs and rabbits. However, it can't take care of snakes and turtles now. Best Friend Pet Care Resort is a very expensive hotel. There are several types of dog rooms, such as rooms with air-conditioners and VIP rooms. There is even a "private dog park". Owners can have a high-quality vacation with their dogs there. The cats are kept in the "Kitty City". The "Kitty City" is very comfortable and clean. Workers there will clean it once a day. Apart from having good equipment, Best Friends Pet Care Resort also have other plans to meet the psychological needs of pets. Workers play music to the pets, walk with them, play with them and even read stories to them before they go to bed. "We aim to make sure that pets will have a fun vacation here, just like their owners at Disney World," said the manager of the resort.
- **Answer:** once a day
- **Prediction:** once a day

Figure 10: Success example 3

Despite reading and learning a long context, the model still locates the right index for the answer. Through the third example, it can be seen that the model can understand the “how often” question. Moreover, the model also understands the relationship between the word “Kitty City” and “it” in the given text.

Failure case analysis

Although the final model achieves a higher performance in both F1 and EM scores, it fails in predicting the answer by the contexts required a higher level of reading comprehension ability. The example above taken from step 30 in the seventh epoch illustrates the failure in giving the correct answer when the model is asked to understand indirect logical associations and prior content related to the question.

Example 4

Context: A family in Guelph, Ontario is spending a year living "in 1982”. They’re doing it so their kids can see what life was like before technology like iPads, computers and even coffee machines was part of everyday life. They have stopped using all technology from their home and are relying on the things people would have used back in the 80s.There is a box at the front door where people can put their cell phones in, while they're visiting the family. Blair McMillan and his wife Morgan want their kids--Trey, 5, and Denton, 2 -- to have a year without technology. Instead of reading e-readers, they are reading books. Instead of using a GPS, they used paper maps. The home the family is living in was built in the 1980s.Even the way they dress and style their hair is from the 80s. The whole idea started when **young Trey** was called to come outside. He didn't because he was busy playing on the iPad. That's why **his dad** thought of the idea of living "in 1982" for a year. The family plans to live like "it's 1982" until April next year. To them, one of the hardest things was giving up their cell phones.

Question: Who made the family have the idea of living "in 1982"?

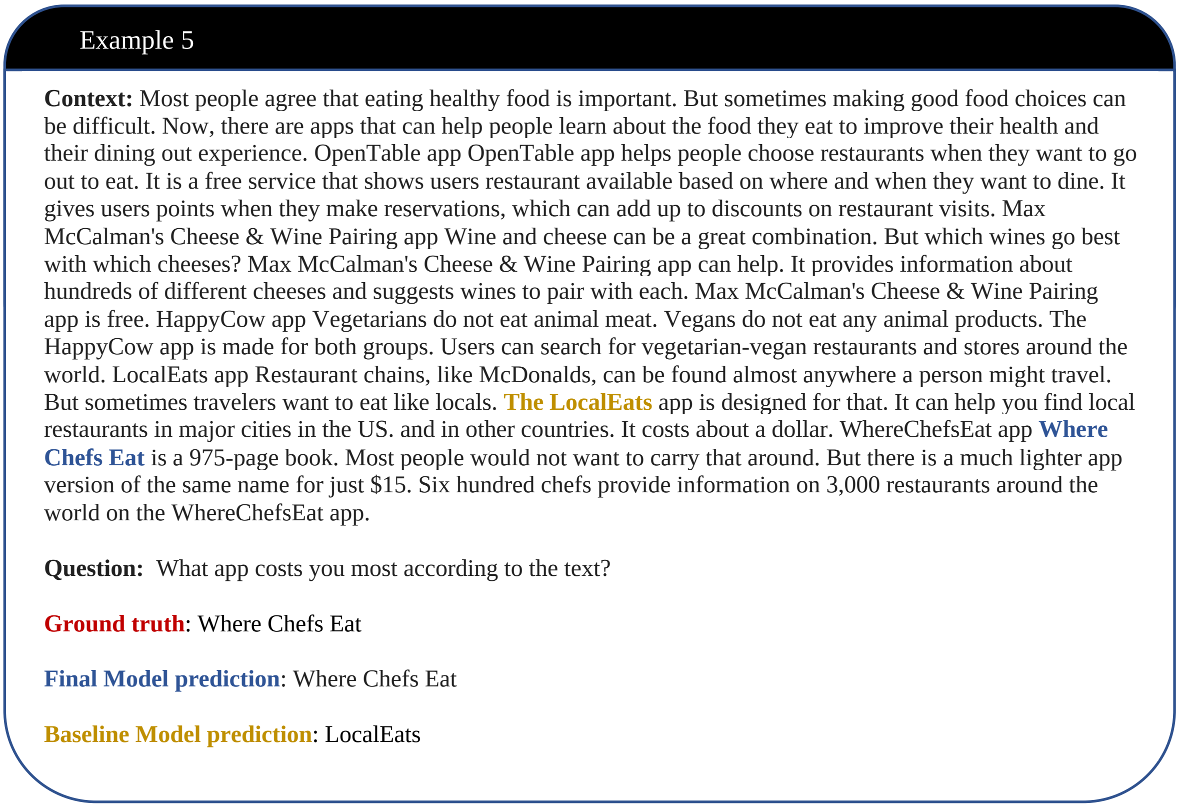
Ground truth: young Trey

Figure 11: Failed example 1

The model seems to be able to recognize the "Who" question with the given content because its answer is among the two people who were involved in the decision even though there are lots of people mentioned in the given passage, which are "young Trey" and "his dad". However, the model was confused between the person who is the reason for the idea of spending a year living "in 1982" and the person who came up with that idea. Specifically, the model fails to understand the phrase "made the family" in the question, so it tends to ignore that phrase. As a result, the model returns the result of a person who thought of the idea instead.

Comparison the final model and baseline

The final model has indicated that it can deal with machine reading comprehension problems required logical reasoning although there is a slight enhancement over the first baseline. The following example illustrates the success of our best model in returning a correct answer based on the given context, which the baseline model does not achieve.



Example 5

Context: Most people agree that eating healthy food is important. But sometimes making good food choices can be difficult. Now, there are apps that can help people learn about the food they eat to improve their health and their dining out experience. OpenTable app OpenTable app helps people choose restaurants when they want to go out to eat. It is a free service that shows users restaurant available based on where and when they want to dine. It gives users points when they make reservations, which can add up to discounts on restaurant visits. Max McCalman's Cheese & Wine Pairing app Wine and cheese can be a great combination. But which wines go best with which cheeses? Max McCalman's Cheese & Wine Pairing app can help. It provides information about hundreds of different cheeses and suggests wines to pair with each. Max McCalman's Cheese & Wine Pairing app is free. HappyCow app Vegetarians do not eat animal meat. Vegans do not eat any animal products. The HappyCow app is made for both groups. Users can search for vegetarian-vegan restaurants and stores around the world. LocalEats app Restaurant chains, like McDonalds, can be found almost anywhere a person might travel. But sometimes travelers want to eat like locals. **The LocalEats** app is designed for that. It can help you find local restaurants in major cities in the US. and in other countries. It costs about a dollar. WhereChefsEat app **Where Chefs Eat** is a 975-page book. Most people would not want to carry that around. But there is a much lighter app version of the same name for just $15. Six hundred chefs provide information on 3,000 restaurants around the world on the WhereChefsEat app.

Question: What app costs you most according to the text?

Ground truth: Where Chefs Eat

Final Model prediction: Where Chefs Eat

Baseline Model prediction: LocalEats

Figure 12: Success example of the final model compared to the baseline.

In the example, the final model has showed its ability to read and learn a long document as well as the relation of between the name of the app and the word "it". However, the model seems to fail in comparing two cost values, which leads to the wrong answer – LocalEats.

The final model further training the first baseline on the out-of-domain and augmented dataset has indicated that task adaptive pretraining technique was not effective for Question Answering task when only implemented on a small set of data. Moreover, an overfitting issue can occur in the training process if applying TAPT to a small dataset [5]. On the other hand, the problem would be mitigated through the data augmentation technique. Without TAPT, back-translation could improve the model's performance. The result suggests a great promise of data augmentation using back-translation that is pretty easy for people to implement.

5.4. Inference process

The inference process includes two steps: pre-processing and post-processing.

Pre-processing:

To prepare the input for the model, the extension takes the question from the user input and parses the HTML DOM tree to retrieve the context. Specifically, it first determines the position of the article content on the page by processing individual nodes of the DOM. Then it removes all of the HTML tags to get the fragments. Finally, the extension connects all the extracted fragments into a context. The web HTLM structure varies from website to website. Therefore, at the moment, our website only supports the https://www.theguardian.com and https://www.bbc.com.

Post-processing:

The model output for a sample consists of two index lists, start index list and end index list. We can use them to find out the actual answer text in the context. The start and end indices should follow some rules, such as the start index should be lower than the end index. Also, we want our extension to return three answers. By doing that, the user has more probability of obtaining the correct information. Picking the best answer is easy since we only need to choose the best start index and the best end index from the output lists. However, it is tricky to pick the second-best: is it the second-best start index with the best end index? Or the best start index with the second-best end index? It is even trickier to pick the third-best answer.

For our project, we choose the three best answers depending on the score of both the start and end indices (there is a score corresponding to each index in the list). After validating them against the rules, we sort all the predictions according to their scores and pick the first three.

6. Application Constraints

At the moment, the project is not product-ready because of few things:

Firstly, the F1 score of our model on the out-of-domain dataset is higher than the baseline, but still **50.82**. With that score, if the web content does not have the same domain as the training set, the model may not provide the correct answer. Also, the answer text should be available in the given content; Otherwise, the model could not find it. In other words, the availability of the answers heavily relies on if there is an answer text in the context since the model cannot generate new text itself.

Secondly, we should have a mechanism to sanitize the user data before re-training the model. In production, it is necessary to cleanse and filter the data to avoid the garbage-in-garbage-out problem. To be more specific, we have to check if the rated answers match the contexts.

Finally, we have not developed the Machine Learning Pipeline to auto-re-train and deploy the models. Since the data in a production environment changes a lot, we need the pipeline to foster the model adaptation; otherwise, the model could not make a correct prediction after the data drift.

III

User guide

1. Installation and Safety Instruction

1.1 Installation

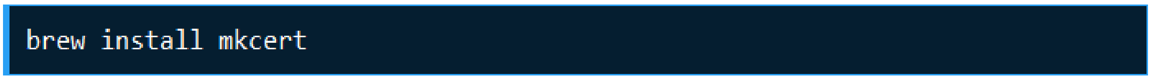
1.1.1 Create Locally Trusted SSL Certificates for using in Google Chrome.

These days, many features require your website to have HTTPS to work, such as Service Workers or some payment gateways will not work if your website does not have HTTPS. So, developing on localhost with HTTP could cause errors. We highly recommend using “mkcert” to solve the above problem simply.

- **For macOS**

1. Turn on your Terminal and make sure that the computer has installed homebrew, otherwise have a look at the document through this URL https://docs.brew.sh/Installation for more information.

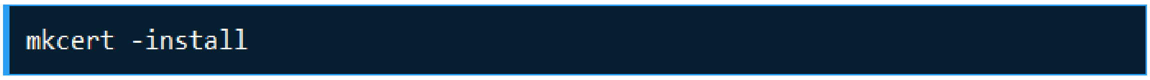
2. Install “mkcert” using homebrew



```
brew install mkcert
```

Figure 13: User guide: command to install mkcert (Mac)

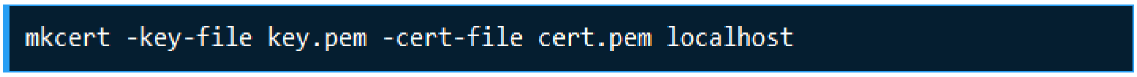
3. Generate and install a local Certificate Authority



```
mkcert -install
```

Figure 14: User guide: command to generate a local certificate authority (Mac)

4. Create a new certificate



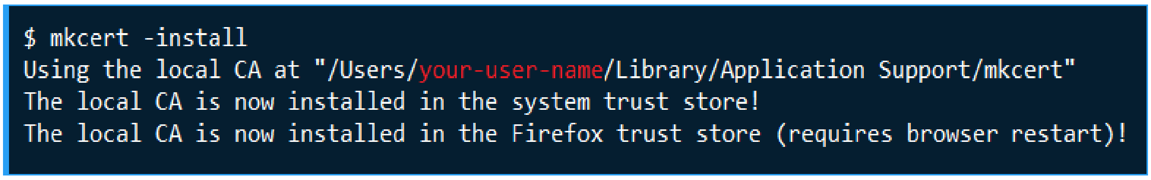
```
mkcert -key-file key.pem -cert-file cert.pem localhost
```

Figure 15: User guide command to create keys (Mac)

- **For Windows:**

1. Install “mkcert”

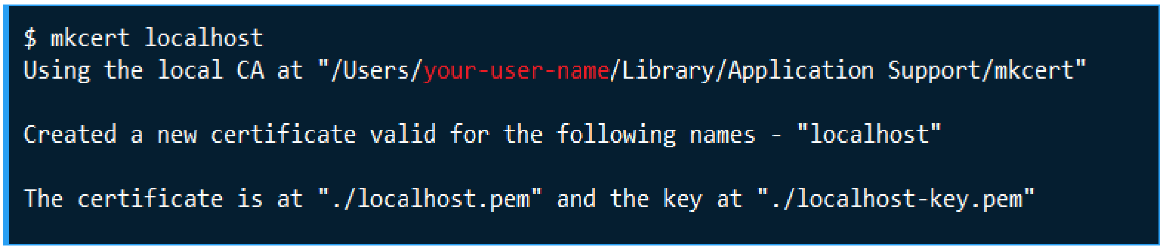
Turn on PowerShell or CMD and install using command:



```
$ mkcert -install
Using the local CA at "/Users/your-user-name/Library/Application Support/mkcert"
The local CA is now installed in the system trust store!
The local CA is now installed in the Firefox trust store (requires browser restart)!
```

Figure 16: User guide: command to install mkcert (Win)

2. Next, make a Certificate Authority (CA). This CA will be saved in local computer.



```
$ mkcert localhost
Using the local CA at "/Users/your-user-name/Library/Application Support/mkcert"

Created a new certificate valid for the following names - "localhost"

The certificate is at "./localhost.pem" and the key at "./localhost-key.pem"
```

Figure 17: User guide: command to make a certificate (Win)

3. After the command finishes running, two files “localhost.pem” and “localhost-key.pem” are created in the directory where the command is being run. Example: *C:\Users\your-user-name*
4. Go to the directory of 2 created keys and save them in the new folder named “keys” Now the path to “localhost.pem” and “localhost-key.pem” is *C:\Users\ your-user-name\keys*
5. Finally, open your Chrome browser, paste and go to *chrome://flags/*
6. Find hashtag *#allow-insecure-localhost* or name “Allow invalid certificates for resources loaded from localhost.” and enable it.

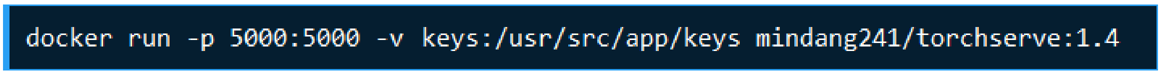
So, the Locally Trusted SSL Certificates has been created successfully to use in Google Chrome.

1.1.2 Set up Docker to run the model.

Docker offers an isolated and disposable environment. So, it is helpful for our developers working with different setups. Besides that, pulling the image can save a lot of time that would otherwise be spent setting up the environment and installing the necessary tools.

- **For macOS:**

1. Install Docker Desktop https://docs.docker.com/docker-for-mac/install/
2. Open Docker Desktop
3. Open Terminal, use this command to start the model.



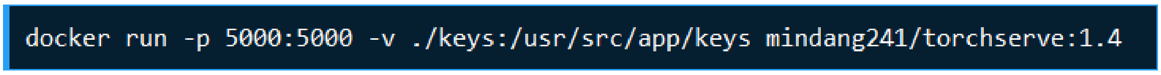
```
docker run -p 5000:5000 -v keys:/usr/src/app/keys mindang241/torchserve:1.4
```

Figure 18: User guide: command to start docker container (Mac)

- **For Windows:**

1. Install Docker https://docs.docker.com/docker-for-windows/install/

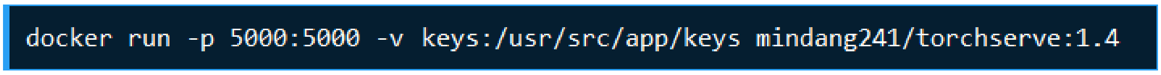
2. Open Docker Desktop
3. Open PowerShell or CMD, use this command to start the model.



```
docker run -p 5000:5000 -v ./keys:/usr/src/app/keys mindang241/torchserve:1.4
```

Figure 19: User guide: command to start docker container (PowerShell)

* If using WSL2 in Windows, please use this command to start the model.



```
docker run -p 5000:5000 -v keys:/usr/src/app/keys mindang241/torchserve:1.4
```

Figure 20: User guide: command to start docker container (WSL2)

1.1.3 Set up the extension in Google Chrome.

HHM’s QA extension currently can be used on many devices without any requirement about the type or version of operating system. Besides that, the extension is developed based on the JavaScript IPA that Chrome provides, so to install and use it in the most stable way, user’s devices must have Google Chrome browser. Now, the way to install is through the source code provided here: https://github.com/lenguyenminhhuy/SEPM-Team24.

Steps to install the extension in Google Chrome:

1. Clone the source code at Github link above.
   - *Some ways to clone a repository from Github: https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/cloning-a-repository*
   - *The local directory which is add for cloning the source code must be saved for future use.*
2. Open Google Chrome browser.
3. Go to extension setting, there are 2 ways to access extension setting in Google Chrome.
   a. Paste and go to *chrome://extensions*.
   b. At the top right, click ⋮ -> More tools -> Extensions.
4. Check “Developer mode” at top right of page.
5. Upload HHM’s QA extension by clicking button “Load unpacked”.

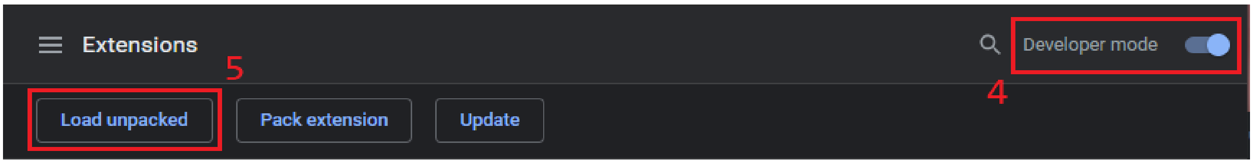
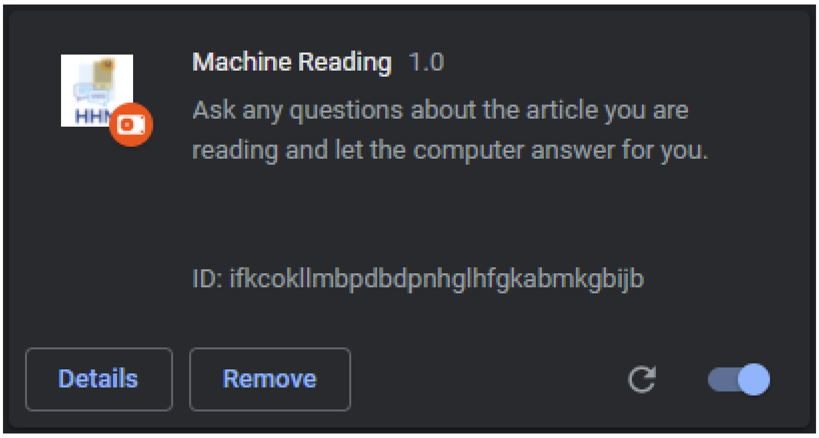


Figure 21: User guide: setup extension step 4&5

6. Enter the local directory, which was saved at step 1.
7. Choose folder “extension” and confirm to load it.



Machine Reading 1.0

Ask any questions about the article you are reading and let the computer answer for you.

ID: ifkcokllmbpdbdpnhglhfgkabmkgbijb

Figure 22: User guide: validate setup

The extension shows like the image above means it is ready to use.

In the next versions, HHM's QA extension will be more complete and is expected to be published in the Chrome Web Store. From there, users can install directly from the store without having to go through the source code.

Steps to install HHM's QA extension from Chrome Web Store:

1. Make sure the device is connected to the internet, then open Google Chrome.
2. Paste and go to https://chrome.google.com/webstore/category/extensions to access extension store in Chrome Web Store.
3. Search for the name "Machine Reading".
4. Click on the HHM's QA extension in the result list.
5. Click button "Add to Chrome" at the right of extension's name.
6. While chrome checks the installation, it may ask for permissions on the websites, click button "Add extension" to agree and complete the installation.
7. HHM's QA extension now appears on *chrome://extensions*.

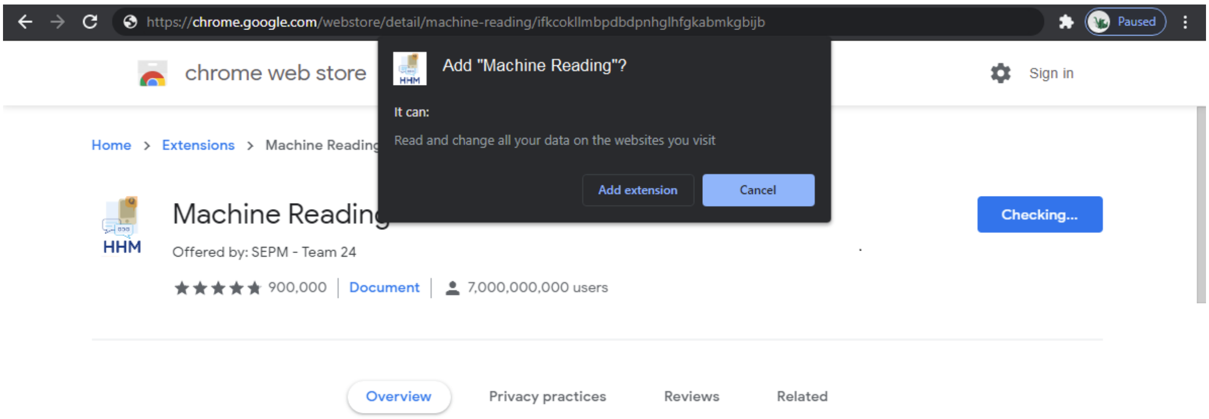
1.2 Safety Instruction

The application does not require user to authenticate; therefore, there is no risk of leaking personal information when using the extension. Moreover, the user feedback does not contain any information related to the user, only the rated answer, question and context.

2. Setup Procedures

HHM's QA extension supports the latest Google Chrome browser version. For the older, the user's browser must be in version 18 or above to be able to install the extension. If user's Google Chrome version is older than 18, please update the browser first.

Allow the extension to read and change all the website's data. At the very first-time users' installation in browser, there will be a dialog asking for usage allowance. Users need to accept this to accept and complete the installation.



Add "Machine Reading"?

It can:

Read and change all your data on the websites you visit

Figure 23: User guide: setup procedures

3. Guides

Home

At every time turn on the extension, the user will always see a simple extension interface like this with name, version, and an input box for entering the question.

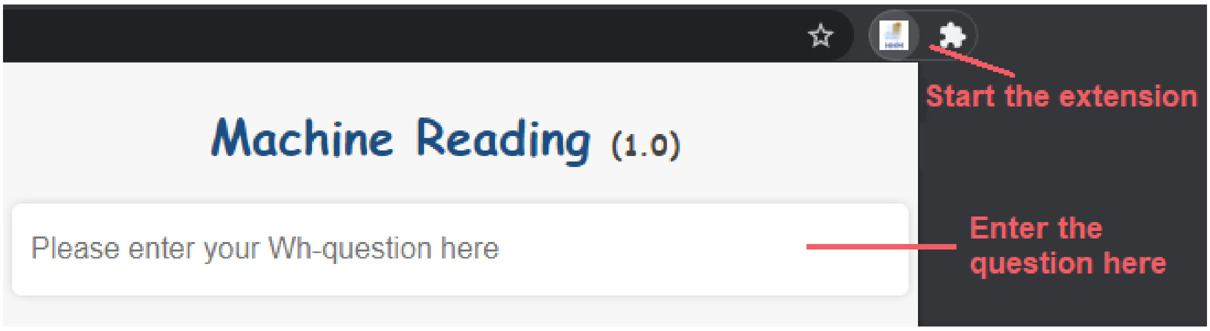
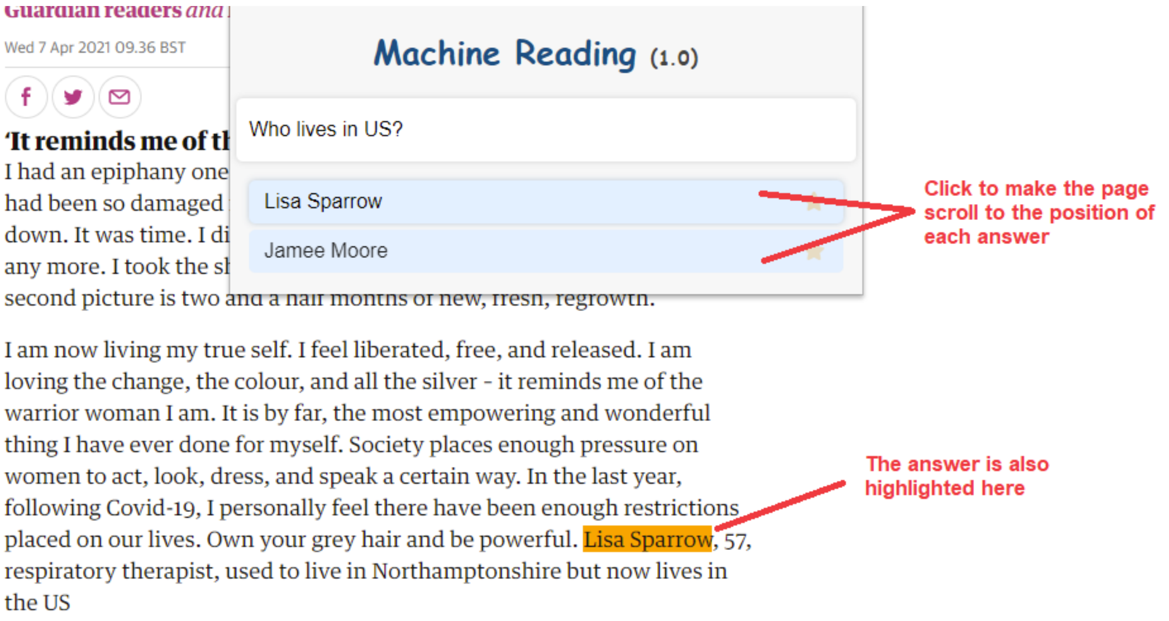


Figure 24: User guide: instruction for asking

Searching

When the user enters a question, the extension will load for a few seconds and display the answers. At the same time, it also scrolls to and highlights the first answer. The user can click on one of the other answers to scroll to and highlight.



Guardian readers and

Wed 7 Apr 2021 09.36 BST

'It reminds me of th

I had an epiphany one

had been so damaged

down. It was time. I di

any more. I took the sl

second picture is two and a half months of new, fresh, regrowth.

I am now living my true self. I feel liberated, free, and released. I am loving the change, the colour, and all the silver - it reminds me of the warrior woman I am. It is by far, the most empowering and wonderful thing I have ever done for myself. Society places enough pressure on women to act, look, dress, and speak a certain way. In the last year, following Covid-19, I personally feel there have been enough restrictions placed on our lives. Own your grey hair and be powerful. Lisa Sparrow, 57, respiratory therapist, used to live in Northamptonshire but now lives in the US

Machine Reading (1.0)

Who lives in US?

Lisa Sparrow ☆

Jamee Moore ☆

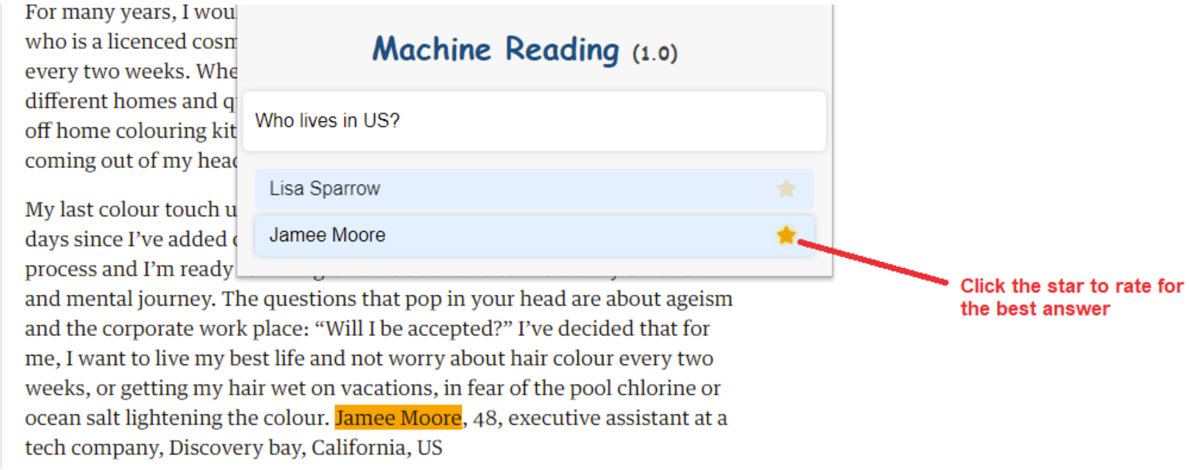
Click to make the page scroll to the position of each answer

The answer is also highlighted here

Figure 25: User guide: instruction for searching

Rating

At the right of each answer's container is a star button used for rating. The user can click on it to rate for the best answer. Yet, this is not required.



For many years, I wou

who is a licenced cosm

every two weeks. Whe

different homes and q

off home colouring kit

coming out of my head

My last colour touch u

days since I've added

process and I'm ready

and mental journey. The questions that pop in your head are about ageism and the corporate work place: "Will I be accepted?" I've decided that for me, I want to live my best life and not worry about hair colour every two weeks, or getting my hair wet on vacations, in fear of the pool chlorine or ocean salt lightening the colour. Jamee Moore, 48, executive assistant at a tech company, Discovery bay, California, US

Machine Reading (1.0)

Who lives in US?

Lisa Sparrow ☆

Jamee Moore ★

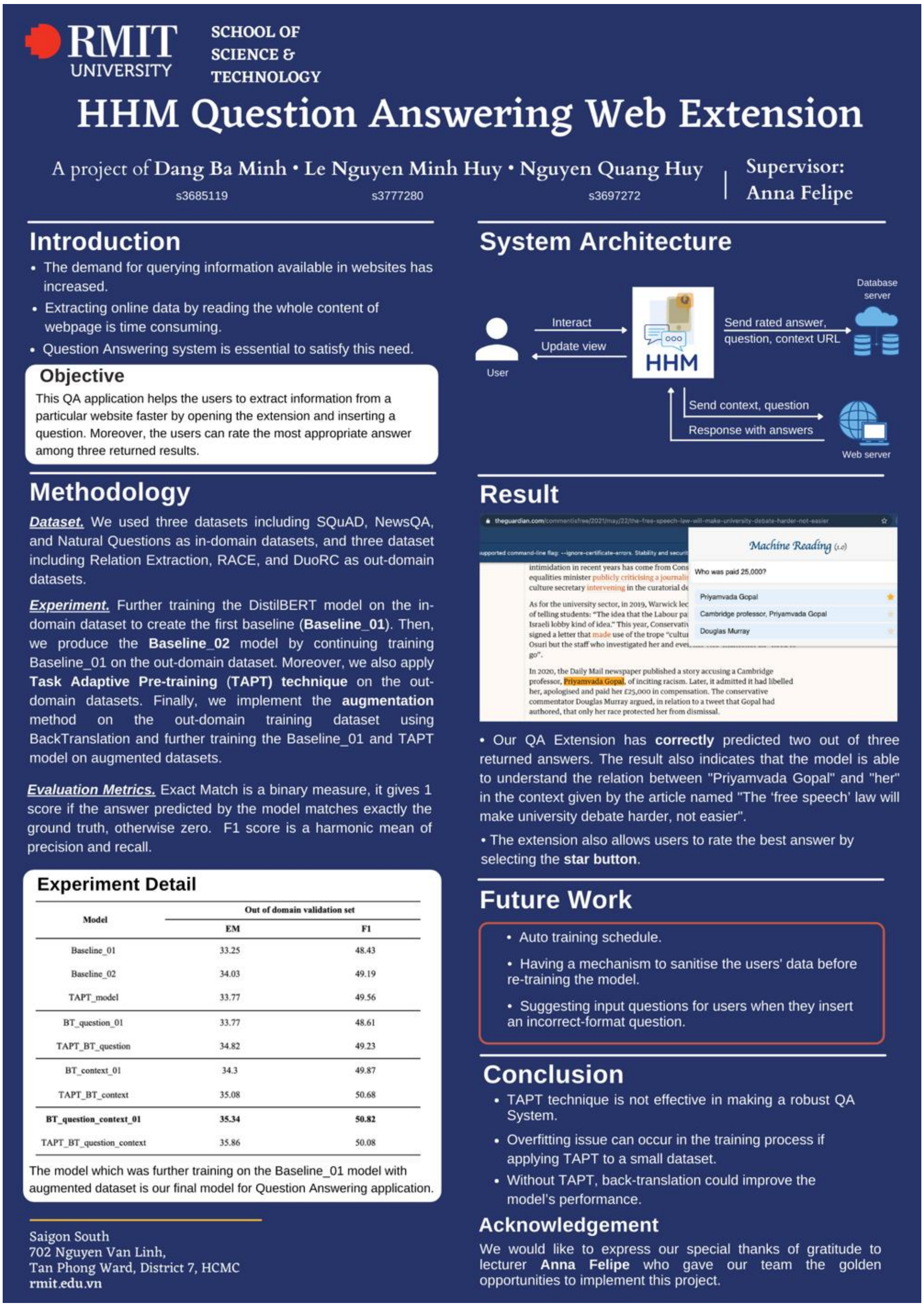
Click the star to rate for the best answer

Figure 26: User guide: instruction for rating

IV

Poster && Brochure

1. Poster



RMIT UNIVERSITY

SCHOOL OF SCIENCE & TECHNOLOGY

HHM Question Answering Web Extension

A project of Dang Ba Minh • Le Nguyen Minh Huy • Nguyen Quang Huy
s3685119 s3777280 s3697272

Supervisor:
Anna Felipe

Introduction

- The demand for querying information available in websites has increased.
- Extracting online data by reading the whole content of webpage is time consuming.
- Question Answering system is essential to satisfy this need.

Objective

This QA application helps the users to extract information from a particular website faster by opening the extension and inserting a question. Moreover, the users can rate the most appropriate answer among three returned results.

Methodology

Dataset. We used three datasets including SQuAD, NewsQA, and Natural Questions as in-domain datasets, and three dataset including Relation Extraction, RACE, and DuoRC as out-domain datasets.

Experiment. Further training the DistilBERT model on the in-domain dataset to create the first baseline (**Baseline_01**). Then, we produce the **Baseline_02** model by continuing training Baseline_01 on the out-domain dataset. Moreover, we also apply **Task Adaptive Pre-training** (**TAPT**) **technique** on the out-domain datasets. Finally, we implement the **augmentation** method on the out-domain training dataset using BackTranslation and further training the Baseline_01 and TAPT model on augmented datasets.

Evaluation Metrics. Exact Match is a binary measure, it gives 1 score if the answer predicted by the model matches exactly the ground truth, otherwise zero. F1 score is a harmonic mean of precision and recall.

Experiment Detail

Model	Out of domain validation set	
	EM	F1
Baseline_01	33.25	48.43
Baseline_02	34.03	49.19
TAPT_model	33.77	49.56
BT_question_01	33.77	48.61
TAPT_BT_question	34.82	49.23
BT_context_01	34.3	49.87
TAPT_BT_context	35.08	50.68
BT_question_context_01	**35.34**	**50.82**
TAPT_BT_question_context	35.86	50.08

The model which was further training on the Baseline_01 model with augmented dataset is our final model for Question Answering application.

Saigon South
702 Nguyen Van Linh,
Tan Phong Ward, District 7, HCMC
rmit.edu.vn

System Architecture

User

Interact

Update view

HHM

Send rated answer, question, context URL

Database server

Send context, question

Response with answers

Web server

Result

theguardian.com/commentisfree/2021/may/22/the-free-speech-law-will-make-university-debate-harder-not-easier

upported command-line flag: --ignore-certificate-errors. Stability and securi

intimidation in recent years has come from Cons
equalities minister publicly criticising a journali
culture secretary intervening in the curatorial de

As for the university sector, in 2019, Warwick lec
of telling students: "The idea that the Labour pa
Israeli lobby kind of idea." This year, Conservativ
signed a letter that made use of the trope "cultu
Osuri but the staff who investigated her and eve
go".

In 2020, the Daily Mail newspaper published a story accusing a Cambridge professor, Priyamvada Gopal, of inciting racism. Later, it admitted it had libelled her, apologised and paid her £25,000 in compensation. The conservative commentator Douglas Murray argued, in relation to a tweet that Gopal had authored, that only her race protected her from dismissal.

Machine Reading (1.0)

Who was paid 25,000?

Priyamvada Gopal ★

Cambridge professor, Priyamvada Gopal ☆

Douglas Murray ☆

- Our QA Extension has **correctly** predicted two out of three returned answers. The result also indicates that the model is able to understand the relation between "Priyamvada Gopal" and "her" in the context given by the article named "The 'free speech' law will make university debate harder, not easier".
- The extension also allows users to rate the best answer by selecting the **star button**.

Future Work

- Auto training schedule.
- Having a mechanism to sanitise the users' data before re-training the model.
- Suggesting input questions for users when they insert an incorrect-format question.

Conclusion

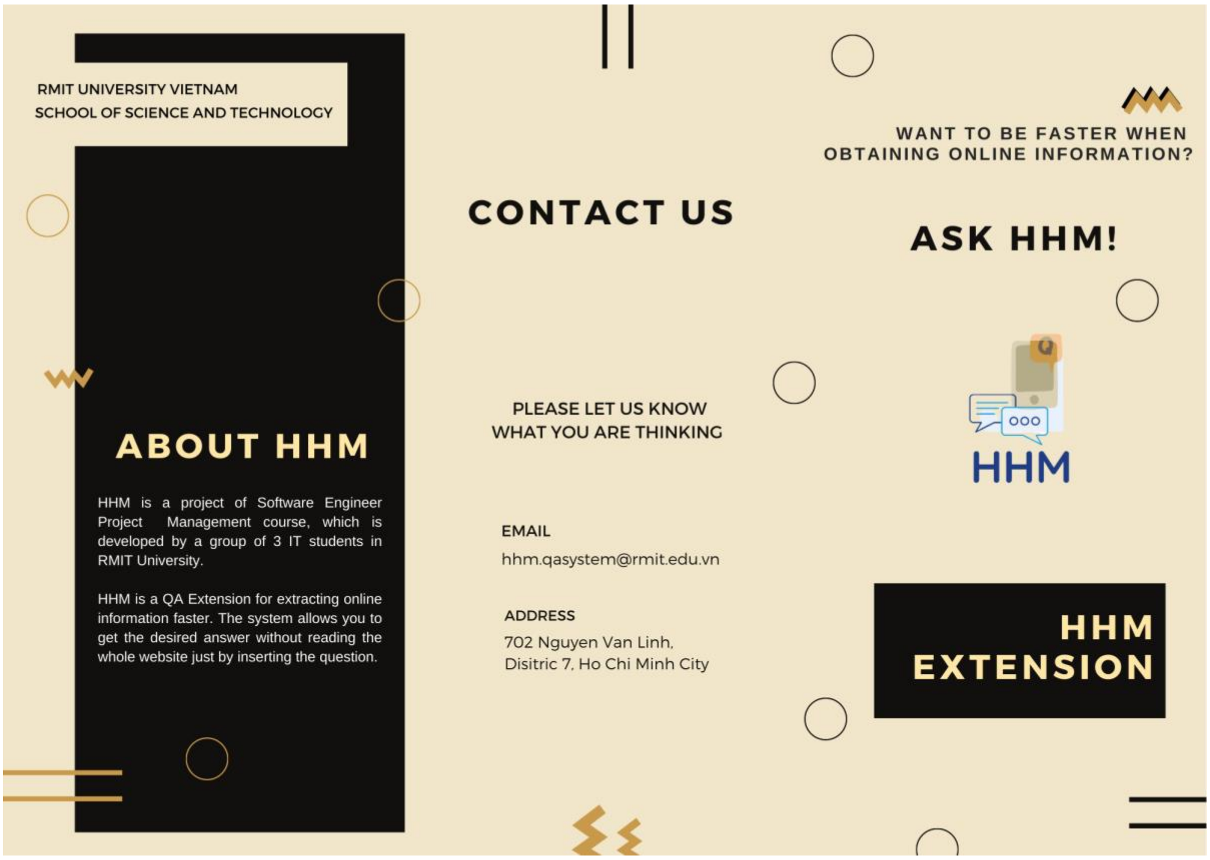
- TAPT technique is not effective in making a robust QA System.
- Overfitting issue can occur in the training process if applying TAPT to a small dataset.
- Without TAPT, back-translation could improve the model's performance.

Acknowledgement

We would like to express our special thanks of gratitude to lecturer **Anna Felipe** who gave our team the golden opportunities to implement this project.

Figure 27: Project Poster

2. Brochure



RMIT UNIVERSITY VIETNAM
SCHOOL OF SCIENCE AND TECHNOLOGY

ABOUT HHM

HHM is a project of Software Engineer Project Management course, which is developed by a group of 3 IT students in RMIT University.

HHM is a QA Extension for extracting online information faster. The system allows you to get the desired answer without reading the whole website just by inserting the question.

CONTACT US

PLEASE LET US KNOW
WHAT YOU ARE THINKING

EMAIL
hhm.qasystem@rmit.edu.vn

ADDRESS
702 Nguyen Van Linh,
Disitric 7, Ho Chi Minh City

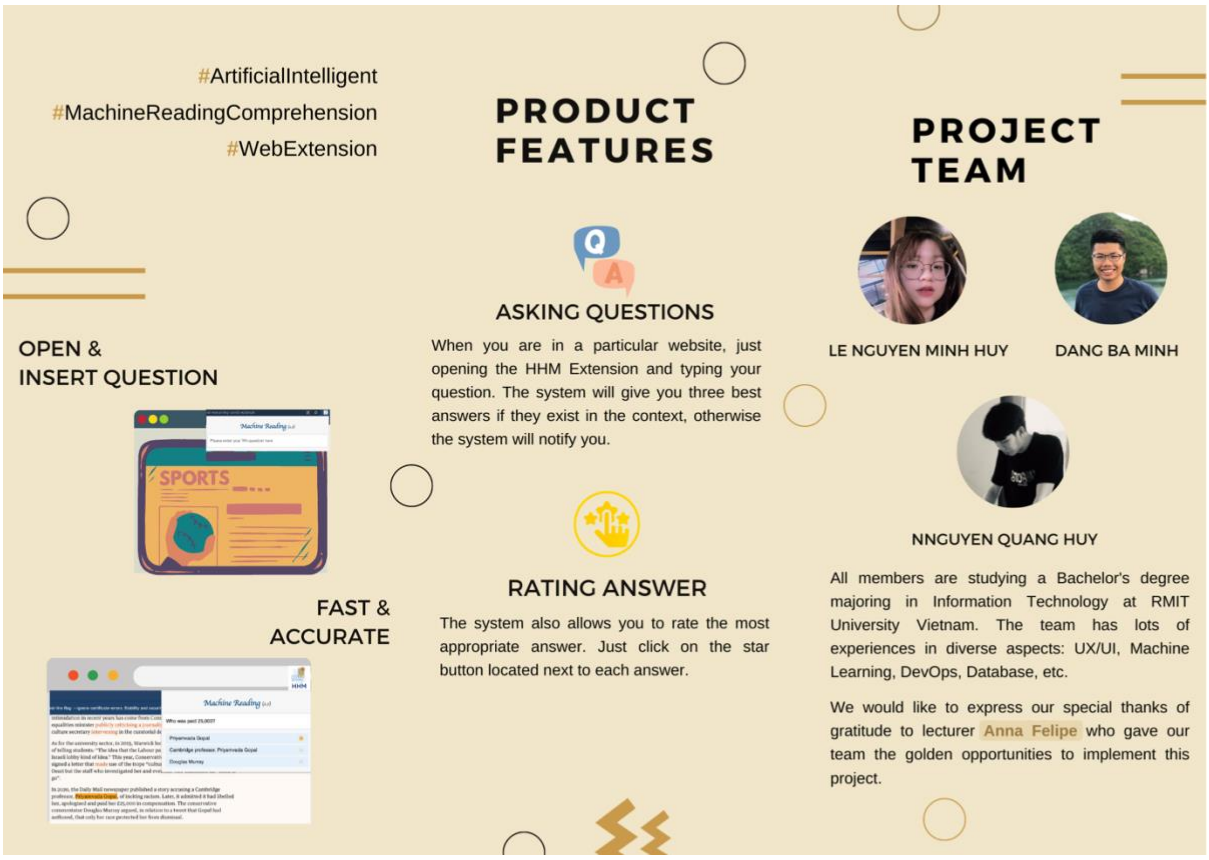
WANT TO BE FASTER WHEN
OBTAINING ONLINE INFORMATION?

ASK HHM!

HHM

HHM
EXTENSION

Figure 28: Outside brochure panels



#ArtificialIntelligent
#MachineReadingComprehension
#WebExtension

OPEN & INSERT QUESTION

FAST & ACCURATE

PRODUCT FEATURES

ASKING QUESTIONS

When you are in a particular website, just opening the HHM Extension and typing your question. The system will give you three best answers if they exist in the context, otherwise the system will notify you.

RATING ANSWER

The system also allows you to rate the most appropriate answer. Just click on the star button located next to each answer.

PROJECT TEAM

LE NGUYEN MINH HUY

DANG BA MINH

NNGUYEN QUANG HUY

All members are studying a Bachelor's degree majoring in Information Technology at RMIT University Vietnam. The team has lots of experiences in diverse aspects: UX/UI, Machine Learning, DevOps, Database, etc.

We would like to express our special thanks of gratitude to lecturer Anna Felipe who gave our team the golden opportunities to implement this project.

Figure 29: Inside brochure panels

V

References

[1] “UML Use Case Diagram: Tutorial with Example,” Guru99, [Online]. Available: https://www.guru99.com/use-case-diagrams-example.html#:~:text=A%20purpose%20of%20use%20case,and%20the%20workflow%20between%20them.. [Accessed 18 5 2021].

[2] “UML Sequence Diagram Tutorial,” Luccidchart. [Online]. [Accessed 18 5 2021].

[3] Yiming Cui, Wanxiang Che , Ting Liu , Bing Qin , Shijin Wang, Guoping Hu, “*Cross-Lingual Machine Reading Comprehension*,” *arXiv:1909.00361*, 1 Sep 2019.

[4] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. *Don't stop pretraining: Adapt language models to domains and tasks*. arXiv preprint arXiv:2004.10964, 2020.

[5] Shayne Longpre, Yi Lu, Zhucheng Tu, and Chris DuBois. *An exploration of data augmentation and sampling techniques for domain-agnostic question answering*. arXiv preprint arXiv:1912.02145, 2019.

[6] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: *Combining local convolution with global self-attention for reading comprehension*. CoRR, abs/1804.09541, 2018.

[7] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. *Squad: 100, 000+ questions for machine comprehension of text.* CoRR, abs/1606.05250, 2016.

[8] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. *Natural questions: a benchmark for question answering research*. In Association for Computational Linguistics (ACL), 2019.

[9] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. *Newsqa: A machine comprehension dataset*. ACL 2017, page 191, 2017.

[10] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. *RACE: Large-scale reading comprehension dataset from examinations*. In EMNLP, 2017.

[11] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. *Zero-shot relation extraction via reading comprehension*. arXiv preprint arXiv:1706.04115, 2017.

[12] Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. *DuoRC: Towards Complex Language Understanding with Paraphrased Reading Comprehension*. In ACL, 2018.