

Generating Custom Knowledge Cards Using Tables on the Web

Zixun Xiang

School of Computer Science

Carleton University

Ottawa, ON, Canada

zixunxiang@cmail.carleton.ca

ABSTRACT

As Knowledge Exploration become a hotspot, many researchers have provided many features and methods for Knowledge Exploration. Some of those knowledge exploration features were relied on knowledge bases and query logs. On the other hand, some other knowledge exploration techniques also associate with the search entity and selected the best results from candidates. The Knowledge Carousels [3], known as one of those kinds of knowledge exploration approach, was proposed to facilitate knowledge exploration of IS-A and HAS-A relationships based on large corpus of tables on the web. In this paper, I propose a framework that is similar to the Knowledge Carousels approach and allows users to customize their knowledge exploration for their needs. My work will make it easier for users who do not have any knowledge exploration experience to generate their own outputs. This including some technical challenges like automatically searching related knowledge, taking user selections, generating human readable outputs. I described how I designed my framework to deal with those three challenges and the experiments demonstrate my framework provided an easy way for users to gain some knowledge exploration outputs as they want.

1 INTRODUCTION

Knowledge Exploration is a way to generate related documents into a query for user. It might sometimes through knowledge panels directly or sometimes uses sections from individual search results. Some search engines (e.g., Bing, Google) use knowledge carousels to help their user get information as effortlessly as possible. A previous work, Knowledge Carousels, also provided a way to use WebTables [9] for Knowledge Exploration. By using those existing tools, knowledge exploration through internet and web tables became easier for every user.

However, auto-generated outputs cannot always fit what the user needs. Search engines and other Knowledge Exploration methods can only generate knowledge card automatically based on some keywords entered by the user. For example, if a user enters a keyword “Kentucky Derby”, search engines will only provide basic information about this keyword and require user to do more research to gain the knowledge they want. On the other hand, Knowledge Carousels framework will provide a downward showing the winners of Kentucky Derby and a sideways representing the famous Triple Crown horse races in the US. Those outputs are better than what search engines can provide but still not always fit the needs from users. If a user only wants to know some specific aspects about this keyword, they need to work on their own and write their own knowledge card by themselves.

As more and more users want to use knowledge exploration as a way for their study and refresh the knowledge in their own

way. Some of them might like to use knowledge card apps like Quizlet and write their own flashcards instead of the outputs from search engines or the Knowledge Carousels framework. However, writing those knowledge cards manually might takes time and not convenience. It requires users to search a lot of materials and organize them on their own. The motivation of this paper is to build a framework that can make it easier for users especially who do not have experience of using Knowledge Exploration tools to generate knowledge cards based on their own needs. Because of this motivation, the new framework I proposed in this paper allows users to generate CSV files by entering the keyword and selecting the attributes they want to learn. The output CSV files they gained from this framework can be automatically loaded into Quizlet as flashcards.

There are some technical challenges for the proposed framework. First, searching for related HTML tables and extracting them from web pages is not easy. Search on the internet always results in a noisy output and it is hard to determine what is needed for the generation. The framework needs to select the HTML tables from the web that related to the user inputs. Second, some information might hide in web pages. This requires the framework allow users to manually add them into the query. Third, generating an easy-to-read file or easy-to-use file is required. This needs the framework format its output when generating it.

By dealing with those challenges above, there are three main contributions of my framework. First, the paper provides an easy way to scrape HTML web tables from selected web pages. This can be used on other knowledge exploration approaches. Second, instead of asking users to search more information for card generating, the framework allows users to enter extra links of the web pages that contain the knowledge they want to include for generation. This helps users to customize their outputs as they want. Third, the output of the framework is CSV file. This means it can be turned into flashcards on Quizlet or used on other purpose.

Section 2 explains how the framework is designed and the workflow of the framework. It also explains each stage in the framework. Section 3 describes how each stages of the framework is implemented and works. Section 4 discusses some limitations of this framework and section 5 provides some experiments that indicates how the frameworks work in real cases. In the end, section 6 introduces some related works and section 7 concludes all this paper.

2 SYSTEM ARCHITECTURE DESIGN

In this section, I provide an overview of how I design the architecture and each stages of the proposed framework. Figure 1 shows the workflow design of the proposed framework. My framework is based on a previous work [3] that using WebTables to generate Knowledge Carousels. Because of this, the proposed framework

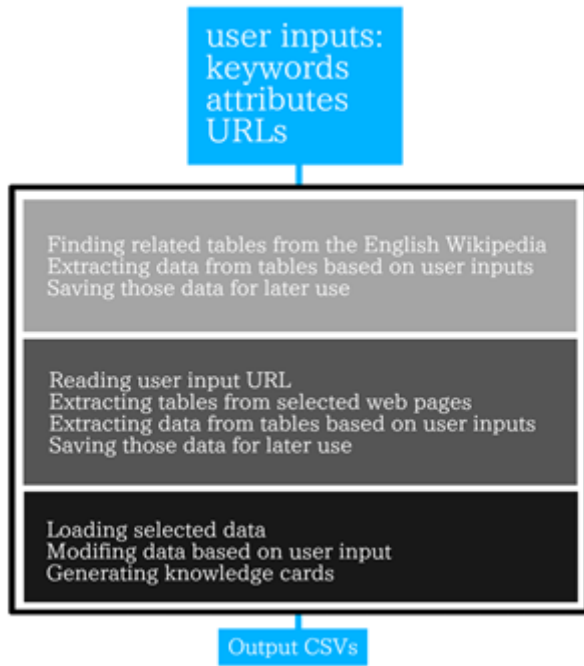


Figure 1: The design workflow of the proposed framework. The system takes user inputs and provides the output as CSV files after three main stages.

focuses more on using HTML tables from the English Wikipedia. To allow users customize the outputs for their own needs, the proposed framework also uses HTML tables from the English web pages that selected by users.

2.1 Scraping Web

The first stage of the proposed framework is to scrape information from the internet based on user input keywords. The information scraped in this stage is the WebTables, refers to HTML tables extracted from Web pages, that related to the keyword.

Figure 2 shows an example of one WebTable for the keyword “CD Projekt Red”. In this extracted HTML table, the rows contain the attributes such as Years, Titles, Platform(s) as well as Notes and the HTML stored the data related to those attributes.

After found the WebTables related to the user input keywords from the English Wikipedia, the system should only scrape the information related to user selected attributes from the extracted WebTables and save those data for later use. For example, is a user entered “CD Projekt Red” as the keyword and chose “Years”, “Titles” as attributes, the framework should scrape and save the data in column “Years” and “Titles” from the WebTables. Figure 3 shows this example.

In conclusion, the first stage should contain three main steps. First, it finds related English Wikipedia page and scraped HTML tables from the page based on user input keyword. Second, it extracts data related to user selected attributes. Finally, it stores the data extracted from the web for the following stages.

2.2 Scraping Selected Web Tables

After getting some basic information and data from the first stage, the second stage allows user to enter URLs that contain the HTML tables they want to use in their knowledge exploration. This means if users think the source from the English Wikipedia is not enough for their knowledge exploration, they can manually add WebTables and attributes they selected by entering the links of the web pages with the HTML tables they want to include.

To make it easier for users to do this, this stage can take all user input URLs and check if those URLs contain HTML tables. If any WebTable is available, the system can extract the related data from those tables based on the keyword and user can select attributes they want from the extracted WebTables. Because of this, users only need to consider whether they need to add more source or not. If they think some external source is necessary for their knowledge exploration, they can add web pages they want to include by simply entering the links to those web pages.

In conclusion, the second stage contains four steps and needs users to make a decision first. The decision from the user decides whether the system run the second stage or not. If the user thinks no external source is needed for the knowledge exploration, the system will jump this stage and proceed to the next stage. On the other hand, if the user adds some links, the system proceeds to the second stage. During this stage, it first takes all user input URLs and check if the web pages contains HTML tables. Next, it extracts related HTML tables from available web pages and asks user to select attributes from those WebTables for their knowledge exploration. Then, it extracts related data from those HTML tables based on the user selected attributes. Finally, it saves those data and stores them with the data from the first stage.

2.3 Generate Output

The third stage for the system is to use the data collected by the previous two stages and generate a CSV file as the output for users.

There are two advantages to have CSV file as the final output. First, CSV file can be loaded as flashcards in Quizlet app. Users can upload their output into Quizlet and get the knowledge cards they want easily. Second, it is better to have structured files as the final output so that users can also use their output in other ways or for other purposes.

In the generation stage, there are three main steps. First, the system loads all stored data from the first and second stages, including the keywords and attributes entered by users. Then, those input keyword and attributes should be modified to generate the title or the name of the output file. Finally, the system should generate a the CSV file that has two columns as the final output for building knowledge cards or other purposes.

Figure 4 shows the output CSV file for a case that the user entered a keyword “CD Projekt Red” and selected two attributes “Years” and “Titles” and added an external URL with attributes “Mode(s)” for the knowledge exploration.

3 IMPLEMENT

In this section, I first explain how the proposed framework do general search with the English Wikipedia and extract expect data from the HTML tables on the Wikipedia pages. Then, I explain my

Year	Title	Platform(s)	Notes
2007	<i>The Witcher</i>	macOS, Microsoft Windows	<i>Enhanced Edition</i> released in 2008
2011	<i>The Witcher 2: Assassins of Kings</i>	Linux, macOS, Microsoft Windows, Xbox 360	<i>Enhanced Edition</i> released in 2012
2014	<i>The Witcher Adventure Game</i>	Android, iOS, macOS, Microsoft Windows	Co-developed with Can Explode ^[64]
	<i>The Witcher Battle Arena</i>	Android, iOS	Co-developed with Fiero Games ^[69]
2015	<i>The Witcher 3: Wild Hunt</i>	Microsoft Windows, PlayStation 4, Xbox One, Nintendo Switch, PlayStation 5, Xbox Series X/S	<i>Expansion pack to The Witcher 3</i>
	<i>The Witcher 3: Wild Hunt – Hearts of Stone</i>		
	<i>The Witcher 3: Wild Hunt – Blood and Wine</i>		
2016	<i>Gwent: The Witcher Card Game</i>	Microsoft Windows, PlayStation 4, Xbox One, iOS, Android	Spinoff of a card game featured in <i>The Witcher 3</i>
2018	<i>Thronebreaker: The Witcher Tales</i>	Microsoft Windows, PlayStation 4, Xbox One, Nintendo Switch, iOS, Android	Standalone single-player game originally planned to be included in <i>Gwent</i> as <i>Gwent: Thronebreaker</i> ^[70]
2020	<i>Cyberpunk 2077</i>	Microsoft Windows, PlayStation 4, Stadia, Xbox One, PlayStation 5, Xbox Series X/S	

Figure 2: A WebTable describing the CDPR games.

Year	Title
2007	<i>The Witcher</i>
2011	<i>The Witcher 2: Assassins of Kings</i>
2014	<i>The Witcher Adventure Game</i>
2015	<i>The Witcher Battle Arena</i>
	<i>The Witcher 3: Wild Hunt</i>
	<i>The Witcher 3: Wild Hunt – Hearts of Stone</i>
2016	<i>The Witcher 3: Wild Hunt – Blood and Wine</i>
2018	<i>Gwent: The Witcher Card Game</i>
	<i>Thronebreaker: The Witcher Tales</i>
2020	<i>Cyberpunk 2077</i>

Figure 3: The saved data based on keyword “CD Project Red” with attributes “Years” and “Titles”

scripts which allow user to enter their selected URLs and choose the attributes from the HTML tables extracted from those web pages. Finally, this section provides an explanation of how the output generation works. Note that, during the implement process, some of the designs are changed to make the system easy to use.

To implement the proposed system, I use three python packages. First, Python’s Wikipedia API [12] is one of the most important external sources I used for the system. The Wikipedia API can help the system retrieve data from Wikipedia automatically. Second, BeautifulSoup [13] is a Python package and can parse HTML documents. This package can create a parse tree for a web pages

The Witcher	2007, Single-player
The Witcher 2: Assassins of Kings	2011, Single-player
The Witcher Adventure Game	2014, Single-player, multiplayer
The Witcher Battle Arena	2015, Multiplayer
The Witcher 3: Wild Hunt	2015, Single-player
The Witcher 3: Wild Hunt – Hearts of Stone	2015, Single-player
The Witcher 3: Wild Hunt – Blood and Wine	2016, Single-player
Gwent: The Witcher Card Game	2018, Single-player, multiplayer
Thronebreaker: The Witcher Tales	2018, Single-player
Cyberpunk 2077	2020, Single-player

Figure 4: The output CSV file example with selected two attributes “Years” and “Titles”

and then programmer can look for components and data they want to extract from the web page by using tags. Finally, Pandas [7] is also an important package that used in this system to modify the extracted tables. This package is an open-source data analysis and manipulation tool that makes the tables modification easier.

Figure 5 shows the final workflow for the proposed system.

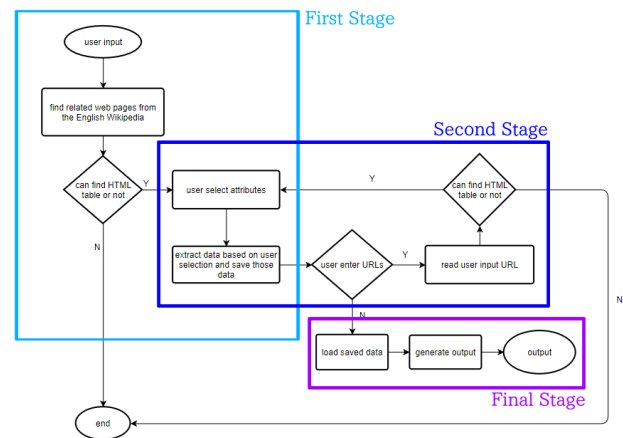


Figure 5: The final workflow of the proposed framework.

Code for the system implementation is available on the GitHub page: <https://github.com/FSZ1X2/COMP5118FinalProject>

3.1 General Search

To get information based on the inputs keyword and attributes, the first thing the system need to do is search the keyword on the

English Wikipedia. To make this process easier, the system uses Python's Wikipedia API[12] to scrape information from the English Wikipedia. It takes user input keyword and search for the Wiki page based on the keyword. After find the related wiki page, the system takes the URL links and extracts the HTML tables from the wiki page by using BeautifulSoup[11] package. After that, the system will print out the extracted HTML table and ask user to select attributes from the WebTable. Then, the system extracts the related columns from the table by using Pandas package[7]. Finally, the system saves all data extracted from the English Wikipedia into a CSV file for later use. Algorithm 1 shows the pseudo-code for this process.

ALGORITHM 1: General Search

```

keyword = user input keyword;
targetURL = wikipedia.page(keyword).url;
soup = targetURL.html.parser;
table = soup.find('table');
if table == null then
    | print "no table can be extracted";
end
if table != null then
    | attributes = user selected attributes;
    | keepcolumns = attributes;
    | tablekeep = panda.readhtml(table)[keepcolumns];
    | tablekeep.saveas('rawdata.csv')
end

```

This process is aimed to scrape some raw data from the English Wikipedia for the final output generation stage. Note that if there is no Wikipedia page related to the keyword or the Wikipedia page related to the user input keyword does not have any HTML table, no data will be collected in this process for the user. If user still want to use the keyword to generate some knowledge cards, they need to input the links related to their subject manually.

3.2 Load User Selected Web

This stage provides a way for users to customize their knowledge cards. Before this stage, user can check the raw data they got from the previous stage and decide whether they want to add more information or not. If users want to include more attributes that are not included in the raw data provided by the previous stage yet, they can enter their selected web pages by simply using the URLs as the input for this stage. The system checks the web pages and extracts HTML tables from those web pages if possible. The system also prints out the extracted tables for users to select the attributes they want for their knowledge cards generation. After user selected attributes, the system loads the stored data from the previous stage and add those selected columns into it. The proposed framework also uses BeautifulSoup package and Pandas package in this process. After all those steps, the system saves all data extracted in this process into a CSV file for later use. Algorithm 2 shows the pseudo-code for this process.

This process is aimed to let users add more information as they expect. One of the advantages of this process is it allows users to

ALGORITHM 2: Load User Selected Web

```

if user entered urls then
    url = user inputs;
    soup = url.html.parser;
    table = soup.find('table');
    if table == null then
        | print "no table can be extracted";
        | require user enter url again
    end
    if table != null then
        | tablePre = read('rawdata.csv');
        | printTable = panda.readhtml(table);
        | attributes = user selected attributes;
        | keepcolumns = attributes;
        | tablekeep = tablePre + printTable[keepcolumns];
        | tablekeep.saveas('rawdata.csv')
    end
end

```

check the HTML tables first and select the attributes they want for their knowledge cards instead of just auto saving all data from the table. Note that the URLs entered by users need to have HTML tables for the data extraction. If there is no HTML table on the web pages, nothing can be extracted from the web pages. To avoid this situation, the proposed system will notice user if there is no HTML table can be extracted.

3.3 Output Generation

From the first two stage, the system should store enough data related to the subject and user selected attributes. The process in this stage is to load those stored data and generate a CSV file for output. The reason for choosing CSV file as the output is that the data in CSV is structured and can be loaded as flashcards in Quizlet app or used in other ways. To generate the final CSV file, the proposed method first loads the data from the previous stages. Then, it compares the attributes and combine columns. For example, if the saved CSV file have three columns: "Title", "Year", "Platform(s)", the generator will automatically modify those columns into two columns: "Title", "Year + Platform(s)". There is not especially preference for the modification. The system will keep the leftist column and combine all the other columns. After the modification, the CSV file should only have two column and the right column will be re-named based on the attributes. Algorithm 3 shows the pseudo-code for this process.

ALGORITHM 3: Output Generator

```

RawTable = read('rawdata.csv');
ColsLeft = RawTable.col[0];
ColRight = RawTable.col[1]+...+RawTable.col[max];
FinalTable[str(ColRight.row[0]) = ColsLeft + ColRight;
FinalTable.saveas('output.csv');

```

Table 1: Statistics for participants

No.	Degree	Area	Experience
1	undergraduate	Math	Y
2	graduate	Computer Science	Y
3	undergraduate	Digital Media	N
4	undergraduate	Digital Media	N
5	undergraduate	Digital Media	N
6	undergraduate	Economy	Y
7	graduate	Computer Science	Y
8	undergraduate	Computer Science	Y
9	undergraduate	Economy	N
10	undergraduate	Economy	N

This process is aimed to provide a file that can be transferred into knowledge cards for users. The output generated in this stage is structured and ready for use. Note that the proposed system also provides the metadata as a CSV file so that there are actually two outputs from the system: one is the final output for knowledge cards and one is a metadata file. If users are not satisfied with the final output, they can still use the metadata CSV file in other analysis tools for other purpose.

4 EXPERIMENTS

In this section, I describe the evaluations I used for validating the proposed framework. There are two points of the evaluations. First, I use the evaluations to check if the new framework can work or not. Second, I examine the quality of the proposed framework. By discussing those two points, the proposed approach is validated as an effective way to generate knowledge cards for users.

4.1 Setup

To investigate if the new framework can work or not and how well it can work, I invite 10 students to evaluate the proposed approach.

Eight of students are undergraduate student. Two of students are graduate student. Three students are in computer science area and have experience with coding knowledge. Five students are in other area and have no experience with coding knowledge. Two students are in other area but confirm that they have experience with coding knowledge. Table 1 shows the full statistics for all participants.

The task for each participant is to select a main subject and write some knowledge cards about the subject they selected. They are asked to first write the knowledge cards manually on Quizlet app and then use the proposed framework to generate the knowledge cards based on their selected subject. There is no limit about the subject selection and the number of knowledge cards.

Before the evaluations, I provide the instruction document for the framework and spend 10 minutes to explain how to use the proposed framework for every participant.

4.2 Metrics

I used three metrics to evaluate the quality of the proposed framework.

First, I evaluate how long each participant need for completing the task by using the proposed approach and by manually writing

the outputs. Those time cost values are compared to evaluate if the proposed framework is more time efficient for users.

Second, I evaluate the output from each participant by using the proposed framework and by manually writing. For the output from the proposed framework, the number of rows is counted. For manually wrote knowledge cards, the number of cards is counted. The evaluation compares those two values and provides if the proposed approach can generate more knowledge cards than manually writing.

Finally, the satisfaction for the output from the proposed framework is collected by the online survey. Each participant rates the output from the proposed approach and the results are collected to calculate the average satisfaction for the proposed approach.

4.3 Workable Test

4.3.1 Methodology. To evaluate the proposed framework, the first step is to check if it can work or not. I write an instruction document and send it to every participant. Before the evaluation, I require all participants to confirm that the framework can run and provide output based on their inputs. To better test the workable of the framework, in the instruction, I provide an output sample with the keyword “CD Project Red” and two attributes “Year” “Title”. If all participants can get the same output by using the same inputs as the sample showed, I consider the proposed framework works properly.

4.3.2 Results. All my participants confirmed that the framework works on their computer properly. This validates that the proposed framework can work for generating outputs for knowledge cards. Note that all of my participants confirmed that they have read my instruction document for the proposed system and understood the workflow of the the proposed framework as well as how it should be work like before they test if this system can work for them.

4.4 Time Cost Evaluation

4.4.1 Methodology. To evaluate the time efficient of the proposed framework, the time cost for manually writing knowledge cards on Quizlet and the time cost for generating knowledge cards by using the proposed framework are collected. For each participant, time cost for manually writing and time cost for using framework are recorded and compared. Less time cost for a method means this method is more time efficient than the other one.

4.4.2 Results. The average time cost for a participant to manually write knowledge cards is 14.9 minutes and the average time cost for a participant to use the framework to generate knowledge cards is 7.3 minutes. On the average, it is clear that using the proposed framework to generate the knowledge cards for a subject cost less time than manually writing the knowledge cards on Quizlet.

There is also an interesting fact. Participants with coding experience or study in computer science area saved more time when using the proposed framework. I think this is because they can better understand how to use the framework and familiar with the interface. Table 2 shows the full results of this evaluation.

Table 2: Time Cost Evaluation Results

No.	Manually write (mins)	By framework (mins)
1	12	5
2	25	6
3	15	5
4	22	8
5	18	7
6	9	5
7	10	7
8	13	6
9	20	15
10	5	9
AVG	14.9	7.3

Table 3: Quality Evaluation Results

No.	Manually write outputs	Rows in framework output
1	6	8
2	15	15
3	10	13
4	10	10
5	9	11
6	5	5
7	7	10
8	10	10
9	15	15
10	5	5
AVG	9.2	10.2

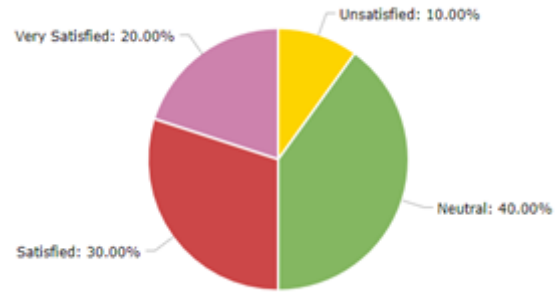
4.5 Output Evaluation

4.5.1 Methodology. To evaluate the quality of the proposed framework, the number of knowledge cards is collected. For each participant’s task, the number of manually wrote knowledge cards and the number of rows in the output CSV file generated by the proposed framework are collect. A large number means the method can provide more knowledge cards and perform better than the other one.

After compared the number of outputs, users’ satisfaction is also another aspect of the quality evaluation. Every participant needs to take a simple survey and rate the outputs from the proposed framework.

4.5.2 Results for quality evaluation. The average number of manually wrote knowledge cards is about 9 cards. On the other hand, the average number of rows in the output file is about 10. This means the proposed framework can generate 10 knowledge cards on average. This result indicates that the proposed method can generate more knowledge cards than manually writing. Note that the number of outputs might be the same for both methods because some keywords with selected attributes might result in exact knowledge cards. Table 3 shows the full results of the evaluation.

4.5.3 Results for quality survey. Most of the participants think there is no difference between the output generated by the proposed framework and their manually wrote knowledge cards. Also, none of the participants think the output generated by the proposed framework is worst than the manually wrote knowledge cards. Two participants think the output generated by the proposed framework is good and better than their manually wrote knowledge cards. Three participants think the output is what they expected and fits what they need. One participant thinks the output is not good enough and the knowledge cards generated by the proposed framework need more detail. Figure 6 provides an overview of the distribution of the survey results.

**Figure 6: Distribution of the satisfaction for the output of the proposed framework**

4.6 Findings

From the results of experiments and evaluations, the proposed framework is validated as a workable one (section 4.3). Also, it takes less time when using the proposed method for generating knowledge cards than manually writing them (section 4.4). Furthermore, the number of knowledge cards generated by the proposed framework is usually equal or more than the number of knowledge cards that users can manually write. The quality of the output from the proposed framework is usually equal to manually wrote knowledge cards or better (section 4.5).

5 LIMITATIONS

In this section, I identify some potential limitations based on the system implementation and experiments.

First, the proposed framework only works with the English Wikipedia and web pages in English. It also only extracts HTML tables from the web pages. So, when using the proposed framework, users need to aware that the web pages they selected need to be in English and the tables on the web pages need to be HTML tables. Also, users need to note that there need to be some HTML tables related to their keyword for knowledge cards generation. If no WebTable can be found for their knowledge generation, no output can be provided for users.

Second, the attributes selected by users need to be existed in the WebTables. If the attributes entered by user is not existed in the extracted WebTables, the system will drop the attribute and no

related data can be collected for generating outputs. Also, for the current version, users need to make sure they typed the attributes they want correctly. This means when user select an attribute, they need to type in exactly same spelling as the WebTable shows.

Third, the system can only use horizontal HTML tables since this kind of tables have are often have higher quality with a better structure. If a selected HTML table has multiple attributes rows with attributes columns, the system might not be able to load this kind of HTML table and extract it from the web page.

Fourth, the evaluation survey indicates that sometimes the output generated by the proposed method have less detail than manually wrote knowledge cards. It is acceptable since the output from the proposed framework is fully based on user inputs and their selected attributes. However, this is still a limitation of the proposed method.

Finally, the quality of user inputs plays an important role in the system process and affects the quality of the output. However, this aspect has not been evaluated in the experiments. It is not clear how the quality of user inputs will influence the results. This problem is one of the future works that need to be evaluated and fixed for the proposed framework.

6 RELATED WORKS

Knowledge Exploration Using Tables on the Web [3] is one of the related works for this paper. This work used tables on the web pages to generate sideways and downward knowledge carousels for knowledge exploration. It leverages the WebTable corpus via sideways and downwards queries and generates the final knowledge carousels for users based on the English Wikipedia. The knowledge carousels technique is used more in mobile interface with browser. The proposed framework is using the same idea to generate knowledge cards for knowledge exploration. However, the proposed method generates a CSV file for users instead of just providing the results.

Web Scraping using Python[10] and BeautifulSoup [11] is another work that related to the proposed method. The proposed method also uses BeautifulSoup package with Pandas package to work on the web scraping process. Works related to Web Scraping using Python and BeautifulSoup explained how to use Python with the BeautifulSoup package to check the HTML parser and scrape information from a web page based on its' HTML tags. This is a good guide for researchers to use information on HTML web pages. Those works provide the basic methods for extracting tables, texts as well as other HTML web components.

Convert HTML tables to CSV files[2][1] is another kind of work related to this paper. In this kind of work, it takes input HTML tables and convert those web tables on the web page into the CSV file format using Python. It can only take HTML tables from the English web pages but the way this work modifies WebTables uses the same idea as the proposed method.

Analysis data by using web scraping technique[14] is also a kind of work related to this paper. this work provided a way to use web scraped data to do some analysis work. It also explained some details about using Python to do the web scraping work. This suggested some methods for the proposed framework to use Python's data analysis packages for knowledge cards generation.

7 CONCLUSIONS

In this paper, I presented a new framework that allows users to build knowledge cards by simply entering the keyword and selected web pages as well as some attributes. I discussed several technical challenges that including searching for related HTML tables and extract them from web pages, reading user inputs and extract web tables from user selected web pages and generating readable CSV file for users to build knowledge cards or use on other purpose. By dealing with those challenges, I built the proposed system and evaluated the system with some experiments.

The experiments validate the work of this paper. It first shows the proposed method can work properly in real world cases. Then, the experiments provide some evidence that indicate the proposed framework is time efficient with good quality. From the survey after the experiments, the quality of the output knowledge cards is validated.

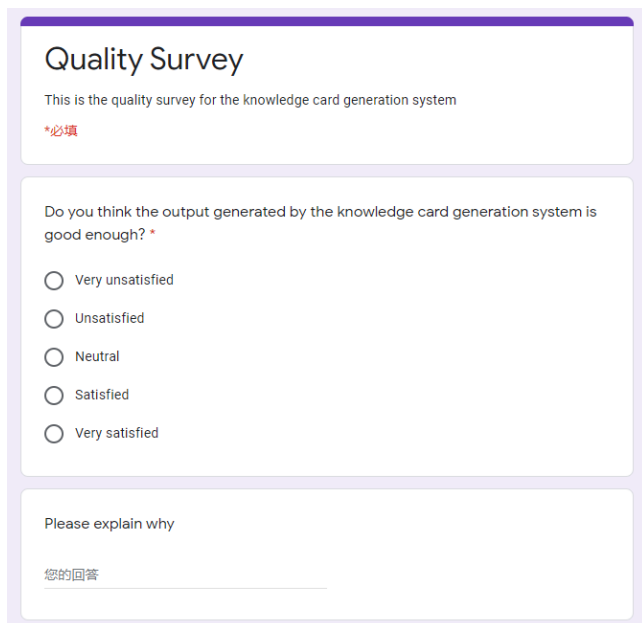
As future work, I plan to improve the system and make other kinds of tables on the web pages works for the proposed framework. The first step is to build a GUI shell for this system. From those evaluations, I found it will be easier for users to select attributes if there is an available GUI interface. This can deal with some limitations I mentioned in the previous section. Second, when improving the proposed system, I also want to reduce the input restrictions. This means even the user inputs might be noisy, the system can still locate the right keywords and attributes for knowledge cards generation. Another future work is to allow language switching for the knowledge cards generation. This means if users want to generate knowledge cards based on their own language, this system can still work for them.

A QUALITY SURVEY

Figure 7 is the screenshot for the quality survey[4] I used to evaluate the user satisfaction of the output generated by the proposed framework.

REFERENCES

- [1] David W Embley, Mukkai Krishnamoorthy, George Nagy, and Sharad Seth. 2011. Factoring web tables. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 253–263.
- [2] David W Embley, Sharad Seth, and George Nagy. 2014. Transforming web tables to a relational database. In *2014 22nd International Conference on Pattern Recognition*. IEEE, 2781–2786.
- [3] Flip Korn You (Will) Wu Cong Yu Fernando Chirigati, Jialu Liu and Hao Zhang. 2016. Knowledge exploration using tables on the web. 1 (Nov. 2016), 193–204. <https://doi.org/10.14778/3021924.3021935>
- [4] Arlene Fink. 2002. *How to ask survey questions*. Vol. 1. Sage.
- [5] Daniel Glez-Peña, Anália Lourenço, Hugo López-Fernández, Miguel Reboiro-Jato, and Florentino Fdez-Riverola. 2014. Web scraping technologies in an API world. *Briefings in bioinformatics* 15, 5 (2014), 788–797.
- [6] Katharine Jarmul and Richard Lawson. 2017. *Python Web Scraping*. Packt Publishing Ltd.
- [7] W McKinney. 2011. pandas: a foundational Python library for data analysis and statistics. 5 (2011). <https://doi.org/10.1109/MCSE.2007.53>
- [8] Wes McKinney et al. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, Vol. 445. Austin, TX, 51–56.
- [9] Daisy Zhe Wang Eugene Wu Michael J. Cafarella, Alon Halevy and Yang Zhang. 2008. WebTables: exploring the power of tables on the web. 2 (Aug. 2008). <https://doi.org/10.14778/1453856.1453916>
- [10] Ryan Mitchell. 2018. Web scraping with Python: Collecting more data from the modern web. 4 (2018). https://doi.org/10.1007/978-3-319-32001-4_483-1
- [11] Leonard Richardson. 2007. BeautifulSoup documentation. (April 2007). <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>.
- [12] Mittal N. Sharma, V. K. 2016. Exploiting Wikipedia API for Hindi-English cross-language information retrieval. 7 (2016), 434–440.



The image shows a web-based survey form titled "Quality Survey". The form is enclosed in a light purple border. At the top, the title "Quality Survey" is in a large, bold, black font. Below the title, a subtitle reads "This is the quality survey for the knowledge card generation system". A red asterisk followed by the Chinese characters "*必填" (required) is positioned below the subtitle. The main question is "Do you think the output generated by the knowledge card generation system is good enough? *". Below this question are five radio button options: "Very unsatisfied", "Unsatisfied", "Neutral", "Satisfied", and "Very satisfied". At the bottom of the form, there is a section titled "Please explain why" with a text input field labeled "您的回答" (Your answer).

Figure 7: The quality survey for evaluate user satisfaction

- [13] Cibambo Masugentwali Steven. 2019. Web Scraping Wikipedia using Python and BeautifulSoup. 3 (Nov. 2019). https://www.researchgate.net/publication/337545583_Web_Scraping_Wikipedia_using_Python_and_BeautifulSoup/citations
- [14] David Mathew Thomas and Sandeep Mathur. 2019. Data analysis by web scraping using python. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 450–454.
- [15] Eloisa Vargiu and Mirko Urru. 2013. Exploiting web scraping in a collaborative filtering-based approach to web advertising. *Artif. Intell. Research* 2, 1 (2013), 44–54.