Spreadsheet Information Retrieval using NLP

Deepak Gami (170101020)
Devaishi Tiwari (170101021)
Soumik Paul (170101066)
Utkarsh Mishra (170101083)

Group 4

1 Problem Statement

Spreadsheet software is one of the most important tool for any organisation. Microsoft's Excel alone has an estimated 750 Million users. They are used to perform data-operations ranging from simple ones like search and filter to very complex ones like pivot tables and linear regression.

However, the plethora of operations provided by these software, makes spreadsheets complicated, especially for novice users. One has to have thorough knowledge about the menus and options within each menu, to be able to execute all set of operations. Apart from the learning complexity, the process of navigating menus is also time consuming.

1.1 Idea Description

We aim to solve this problem by creating an intelligent layer on top of a spreadsheet, which provides an alternative interaction method via Natural Language. Hence, instead of having to navigate menus, a user can simply describe his/her query in natural language and get the required results faster.

An open-sourced NLP enabled powerful spreadsheet can substantially democratize usage of complex statistical operations over spreadsheets.

1.2 Advantages

The advantages of our work could be:

- Faster & easier workflows for existing users of spreadsheet
- Reduced knowledge barrier to perform statistical analysis over spreadsheets
- By combining this with speech processing, spreadsheet software can be made usable by people with disabilities who have difficulties in handling mouse and keyboard

2 Major Challenges

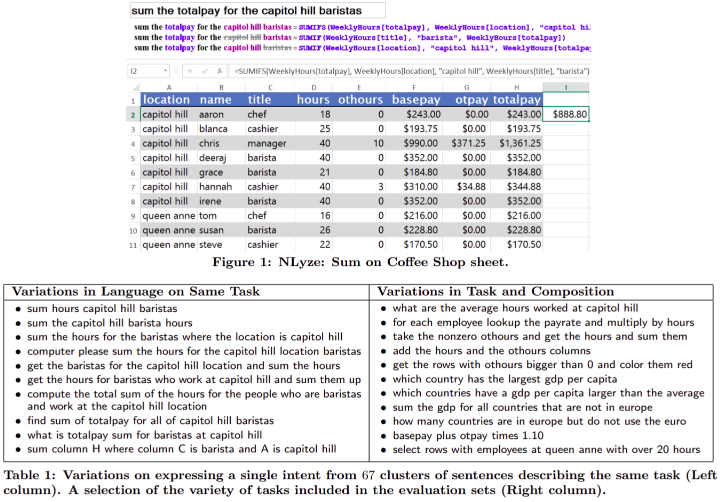
As discussed in Microsoft's Research Paper [GM14], the complexity is enhanced due to two reasons:

1. Variations in Language on Same Task Variations in Task and Composition
2. Variations in Task and Composition

Apart from this, all queries are highly contextual with multiple references to column names, table name and/or data entries. Our methods have to take this into consideration.

3 Existing Works

- NLyze [GM14]: Researchers at Microsoft built an interface using a Programming by Natural Language (PBNL) methodology by developing a Domain specific language(DSL) which expresses the desired task and translation algorithm for converting natural language specifications into a ranked set of likely programs in the DSL.
- NLP-SIR[FMM09] : This research has outlined the effectiveness and efficiency of spreadsheet information retrieval using a Natural language interface which is uses NLP techniques to interpret the commands.



sum the totalpay for the capitol hill baristas

sum the totalpay for the capitol hill baristas ≈ SUMIFS(WeeklyHours[totalpay], WeeklyHours[location], "capitol hi
sum the totalpay for the ~~capitol hill~~ baristas ≈ SUMIF(WeeklyHours[title], "barista", WeeklyHours[totalpay])
sum the totalpay for the capitol hill ~~baristas~~ ≈ SUMIF(WeeklyHours[location], "capitol hill", WeeklyHours[totalpa

I2 =SUMIFS(WeeklyHours[totalpay], WeeklyHours[location], "capitol hill", WeeklyHours[title], "barista")

	A	B	C	D	E	F	G	H	I
1	location	name	title	hours	othours	basepay	otpay	totalpay	
2	capitol hill	aaron	chef	18	0	$243.00	$0.00	$243.00	$888.80
3	capitol hill	blanca	cashier	25	0	$193.75	$0.00	$193.75	
4	capitol hill	chris	manager	40	10	$990.00	$371.25	$1,361.25	
5	capitol hill	deeraj	barista	40	0	$352.00	$0.00	$352.00	
6	capitol hill	grace	barista	21	0	$184.80	$0.00	$184.80	
7	capitol hill	hannah	cashier	40	3	$310.00	$34.88	$344.88	
8	capitol hill	irene	barista	40	0	$352.00	$0.00	$352.00	
9	queen anne	tom	chef	16	0	$216.00	$0.00	$216.00	
10	queen anne	susan	barista	26	0	$228.80	$0.00	$228.80	
11	queen anne	steve	cashier	22	0	$170.50	$0.00	$170.50	

Figure 1: NLyze: Sum on Coffee Shop sheet.

Variations in Language on Same Task	Variations in Task and Composition
• sum hours capitol hill baristas • sum the capitol hill barista hours • sum the hours for the baristas where the location is capitol hill • computer please sum the hours for the capitol hill location baristas • get the baristas for the capitol hill location and sum the hours • get the hours for baristas who work at capitol hill and sum them up • compute the total sum of the hours for the people who are baristas and work at the capitol hill location • find sum of totalpay for all of capitol hill baristas • what is totalpay sum for baristas at capitol hill • sum column H where column C is barista and A is capitol hill	• what are the average hours worked at capitol hill • for each employee lookup the payrate and multiply by hours • take the nonzero othours and get the hours and sum them • add the hours and the othours columns • get the rows with othours bigger than 0 and color them red • which country has the largest gdp per capita • which countries have a gdp per capita larger than the average • sum the gdp for all countries that are not in europe • how many countries are in europe but do not use the euro • basepay plus otpay times 1.10 • select rows with employees at queen anne with over 20 hours

Table 1: Variations on expressing a single intent from 67 clusters of sentences describing the same task (Left column). A selection of the variety of tasks included in the evaluation sets (Right column).

Figure 1: Examples of sample sentences that showcase the variations

4 Proposed Direction

After some preliminary readings, we have designed the following method:

1. Generate a set of prototypical Natural Language Queries to perform the operations. These NLQ's have to be abstract, so that they can be replaced with contextual parameters later. Eg: For operation "FILTER", a prototypical Natural Language Query can "List all <table_name> that have <col_name> greater than <data_entry>". Terms in <> will be replaced as per the queries context. Let the set of NLQ's be S. Our set will be tuple with each entry being {NLQ, operation}.

2. Now given a query(say q), get the contextual information and replace all the abstractions (ie <> terms) in S to get S_q

3. Next, find the NLQ from S_q which is semantically closest to q. This will tell us what operation needs to be performed.

4. Given the operation & contextual information, we can execute and return the results.

This is just the basic idea, and we need to test it before finalizing. Our next step would be to pick a small subset of operations (1 or 2 simple ones like FILTER, ADD) and test this method. After we have a good method to perform basic operations, we will try to enhance it to handle complex operations (eg: ADD + FILTER).

References

[FMM09] Derek Flood, Kevin McDaid, and Fergal McCaffery. "NLP-SIR: A Natural Language Approach for Spreadsheet Information Retrieval". In: *CoRR* abs/0908.1193 (2009). arXiv: 0908.1193. http://arxiv.org/abs/0908.1193.

[GM14] Sumit Gulwani and Mark Marron. "NLyze: Interactive Programming by Natural Language for SpreadSheet Data Analysis and Manipulation". In: *SIGMOD'14, June 22-27, 2014, Snowbird, UT, USA*. June 2014. https://www.microsoft.com/en-us/research/publication/nlyze-interactive-programming-natural-language-spreadsheet-data-analysis-manipulation/.