The sentiment and need analysis of type 1 and type 2 diabetes

Group 4

Team members

- CHAN Kiu Nga, Sandy (56219011)
- CHAN Tsz Ngai, Matthew (56213172)
- Ma Wing Kin, Paul (56243594)
- NG Sin Yung, Cindy (56213971)

Table of contents

01

Introduction & Data extraction

02

Data processing

03

Methods & results

04

Conclusion

01
Introduction & Data extraction

The type of diabetes

Type 1

- Cause by an **autoimmune reaction**. This **destroys the cells in the pancreas that make insulin .**

Type 1.5 (LADA)

- **Between** from diabetes type 1 and diabetes type 2

Type 2

- Cause by **pancreas does not produce enough insulin.**
- **Most common** diabetes type

But what are the *needs* and *sentiment* of these diabetes patients?

Diabetes forum

To understand the **needs and sentiments** , we extracted information from the 2 diabetes forums

Because they are...

✓ **Can be extracted without permissions**
✓ **Divided the discussions into group for 3 diabetes types -> fair comparison for each diabetes' type popularuty in the forums**

Websites	Type 1 forum	Type 1.5 forum	Type 2 forum
Diabetes Daily Forum	✓	✓	✓
Diabetes.co.uk	✓	✓	✓

Extraction method

1. Import required package

In [3]:
```
#import all the required packages
#web scrapping
import requests
from bs4 import BeautifulSoup as bs

#serve as complement`
import time
import pandas as pd
```

Extraction method

2. Define the function for extracting the maximum possible number of pages in a fourm's / discussion topic

In [4]:
```
def get_NumberOfPages (url):
    #use requests.get to get the content
    req = requests.get(url)
    #use BeautifulSoup of the text
    soup = bs(req.content, 'html.parser')
    #find the hyperlinks in <ul> ("class" : "pageNav-main" })
    #navs is a list
    navs = soup.find("ul", { "class" : "pageNav-main" })
    #num_of_pages is 1 when navs cannot find anything (no additional pages)
    page_length=1
    if (navs):   #if the navs exist, which means there are one more pages
        navs=navs.find_all("li", recursive=False)
        #navs is a list that show how many page buttons in the same page, since the website will
        #show first several pages , then the number of last page,
        #so if we retrieve the last element in navs and convert it to text then int
        #then we can know how many number of pages need to scrape
        page_length=int(navs[len(navs)-1].text)

    return page_length
```

In [5]:
```
target_forum_url="https://www.diabetesdaily.com/forum/forums/type-1-diabetes.9/"
total_pages=get_NumberOfPages (target_forum_url)
print("Total number of pages in disscussion forum:",total_pages)
```

```
Total number of pages in disscussion forum: 915
```

Extraction method

3. A user defined function for extracting the data

(plug the page number to the function)

```
In [6]: #Build up the lists to hold the data

        discussion_topics=[]
        NoOfTexts_discussion_topics=[]
        Texts=[]
```

```
#Prevent none type cannot be decomposed bug
#Define a big function to do all the hold the data
def getData (page_number):
    #declare the global lists into function
    global discussion_topics
    global NoOfTexts_discussion_topics
    global Texts

    topic_diag=[] #the number of dialogues of each topic

    forum_url="https://www.diabetesdaily.com/forum/forums/type-1-diabetes.9/page-"+str(page_number)

    #extract all the discussion fourm links from each page
    source_html = requests.get(forum_url)

    topic_pagelinks = [
    f'https://www.diabetesdaily.com{a["href"]}' for a
    in bs(source_html.content, "html.parser").select(".structItem-title a")]
    #for j in topic_pagelinks:print(type(j))
    for link in topic_pagelinks:

        #form the information
        topic_url=link
        topic_req=requests.get(link)
        topic_sp=bs(topic_req.text,'lxml')

        #first we extract the TITLE
        topic_title=topic_sp.find("h1",{"class":"p-title-value"}).text
        discussion_topics.append(topic_title)
        #print(topic_title)

        #In each discussion topic, we may have more than one page

        #so the first step is, we know how many pages are there
        topic_pages_no=get_NumberOfPages(link)
        #print(topic_pages_no)

        topic_content=[] #the contents (text) in each topic
        number_of_diaglogues_per_topic=0
        #use for loop to extract ALL content also the number of dialogues
        for page_num in list(range(1,topic_pages_no+1)):
                topic_page_url=topic_url
                topic_page_url=topic_page_url+"page-"+str(page_num)
                #print(topic_page_url)
                topic_page_r=requests.get(topic_page_url)
                topic_page_sp=bs(topic_page_r.text,'lxml')
                topic_list_content_per_page=topic_page_sp.find_all("div",{"class":"bbWrapper"})
                for element in topic_list_content_per_page:
                    remove=element.find("div",{"class":"bbCodeBlock-content"})
                    if (remove!=None):
                        element.find("blockquote").decompose()
                    else:
                        continue

                topic_list_content_per_page=[i.text for i in topic_list_content_per_page]
                topic_content.extend(topic_list_content_per_page)
        #print(len(topic_content))
        number_of_diaglogues_per_topic+=len(topic_content)


        #after extracted all topic's all content , append it into the global list:
        #why append ? because we can explode the list inside topic's content by each disscussion forum's topic after the dataframe is formed
        Texts.append(topic_content)

        #add the number of dialogues into the global list
        NoOfTexts_discussion_topics.append(number_of_diaglogues_per_topic)
```

Extraction method

4. Information exrtraction (Normal Computing)
(Need self-checking
As decompose texts error is easily occurred in long-time extraction)

```
In [10]:
start=time.time()
for i in list(range(1,total_pages+1)):
    getData(i)
end=time.time()
exe_time=end-start
print("Parallel computing extraction total processing time: ",exe_time," s")
```

```
      1 start=time.time()
      2 for i in list(range(1,total_pages+1)):
----> 3     getData(i)
      4 end=time.time()
      5 exe_time=end-start

Input In [7], in getData(page_number)
     25 topic_sp=bs(topic_req.text,'lxml')
     27 #first we extract the TITLE
---> 28 topic_title=topic_sp.find("h1",{"class":"p-title-value"}).text
     29 discussion_topics.append(topic_title)
     30 #print(topic_title)
     31
     32 #In each discussion topic, we may have more than one page
     33
     34 #so the first step is, we know how many pages are there

AttributeError: 'NoneType' object has no attribute 'text'
```

Extraction method

When decompose error, small function is needed .
-> adding the disscussion topics by its url

```
#stop at p.343 'Humalog v Apidra', so next I would start the loop at p.344
#p.343 remaining topics -> extract MANNUALLY

#I defined a per topic function to add to the global variable if failed
#'Test Results And What I Was Told' #NoOfTexts_discussion_topics need to add ourselves
def getData_topic(specific_url):
        global discussion_topics
        global Texts

        specific_get=requests.get(specific_url)
        specific_sp=bs(specific_get.text,'lxml')
        specific_number_of_texts=0
        specific_number_of_forum_pages=get_NumberOfPages (specific_url)

        #get the title
        specific_topic_title=specific_sp.find("h1",{"class":"p-title-value"}).text
        discussion_topics.append(specific_topic_title)

        specific_topic_content=[]

        for page_num in list(range(1,specific_number_of_forum_pages+1)):
                specific_topic_page_url=specific_url
                specific_topic_page_url=specific_topic_page_url+"page-"+str(page_num)
                #print(topic_page_url)
                specific_topic_page_r=requests.get(specific_topic_page_url)
                specific_topic_page_sp=bs(specific_topic_page_r.text,'lxml')
                specific_topic_list_content_per_page=specific_topic_page_sp.find_all("div",{"class":"bbWrapper"})
                for element in specific_topic_list_content_per_page:
                    specific_remove=element.find("div",{"class":"bbCodeBlock-content"})
                    if (specific_remove!=None):
                        element.find("blockquote").decompose()
                    else :continue
                specific_topic_list_content_per_page=[i.text for i in specific_topic_list_content_per_page]
                specific_topic_content.extend(specific_topic_list_content_per_page)

        Texts.append(specific_topic_content)
```

Extraction method

Self- checking code for checking process go wrong correct order is important for us in analysis.

Self-checking for last elements , put it at back

In [73]:
```
#the length of lists
#don't use [] again! CRASHED MY MIND!!
print("Length of discussion_topics: ",len(discussion_topics))
print("-----------------------------------------------------------")
print("Length of NoOfTexts_discussion_topics: ",len(NoOfTexts_discussion_topics))
print("-----------------------------------------------------------")
print("Length ofTexts: ",len(Texts))
print("-----------------------------------------------------------")
```

```
Length of discussion_topics:  9132
-----------------------------------------------------------
Length of NoOfTexts_discussion_topics:  9132
-----------------------------------------------------------
Length ofTexts:  9132
-----------------------------------------------------------
```

In [83]:
```
#the last element of list

print("Latest discussion topics ",(discussion_topics)[-1])
print("-----------------------------------------------------------")
print("Length of NoOfTexts_discussion_topics: ",(NoOfTexts_discussion_topics)[-1])
print("-----------------------------------------------------------")
print("Texts in latest topic: ",((Texts)[-1][NoOfTexts_discussion_topics[-1]-1]))
print("-----------------------------------------------------------")
print("Length of Texts in latest topic: ",len((Texts)[-1]))
print("-----------------------------------------------------------")
```

```
Latest discussion topics  Taking Advantage of Diabetics
-----------------------------------------------------------
Length of NoOfTexts_discussion_topics:  4
-----------------------------------------------------------
Texts in latest topic:  http://msnbc.msn.com/id/10777506/
-----------------------------------------------------------
Length of Texts in latest topic:  4
-----------------------------------------------------------
```

Extraction method

5. Add the result of lists to pd.DataFrame

In [85]:
```
# group extraction into the dataset
data1=pd.DataFrame(columns=["topic","number of text"])
data1["topic"]=discussion_topics
data1["number of text"]=NoOfTexts_discussion_topics
data1
```

Out[85]:

	topic	number of text
0	Learning Center - Type 1 Diabetes	6
1	Had a Friend with type one	65
2	Seeking Diary Study Participants (children wit...	1
3	An insulin users joke	2
4	t:connect training quiz	3
...	...	...
9127	have you got a spotty head	1

In [86]:
```
data2=pd.DataFrame(columns=["topic","text"])
data2["topic"]=discussion_topics
data2["text"]=Texts
data2=data2.explode("text") #explode the list in text -> discrete data
data2
```

Out[86]:

	topic	text
0	Learning Center - Type 1 Diabetes	For up to date information about Type 1 Diabet...
0	Learning Center - Type 1 Diabetes	I have a question about insulin antibody testi...
0	Learning Center - Type 1 Diabetes	In general, a reference range for any given te...
0	Learning Center - Type 1 Diabetes	Thank you for the reply. I think you must be ...
0	Learning Center - Type 1 Diabetes	Medtronics insulin pump with MiniMed Quick -se...
...	...	...

```
#export both of them to csv
data1.to_csv("DiabetesDailyForum_Type1_CommentCountPerTopic.csv")

data2.to_csv("DiabetesDailyForum_Type1_CommentPerTopic.csv")
```

Extraction method

- Why no parallel computing?
- It's fast
- But the 3 lists have dependencies -> the order is **SHUFFLED** among lists ->not good for analysis
- Comparison

- 3 pages extraction comparison

Out[92]:

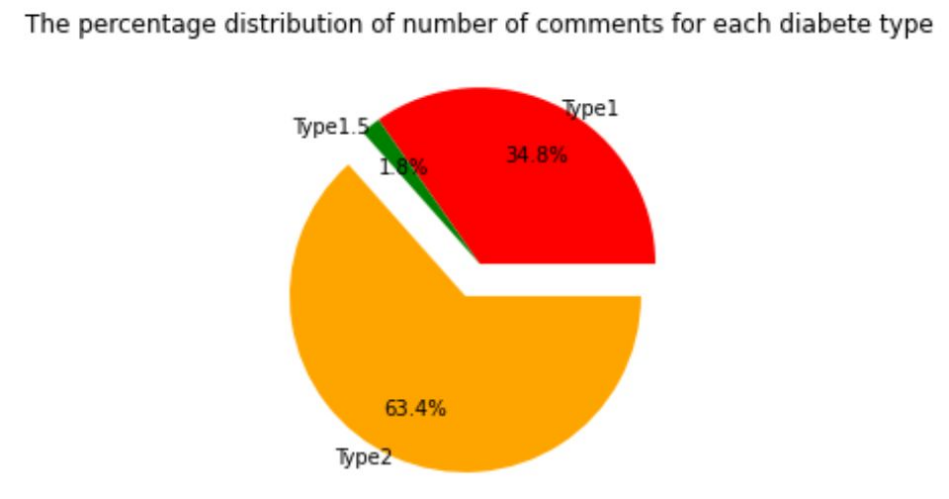
	Normal computing	Parallel computing (n_jobs=16)
Time (second)	67.624082	34.264024
Data Accuracy	Accurate	Non-accurate

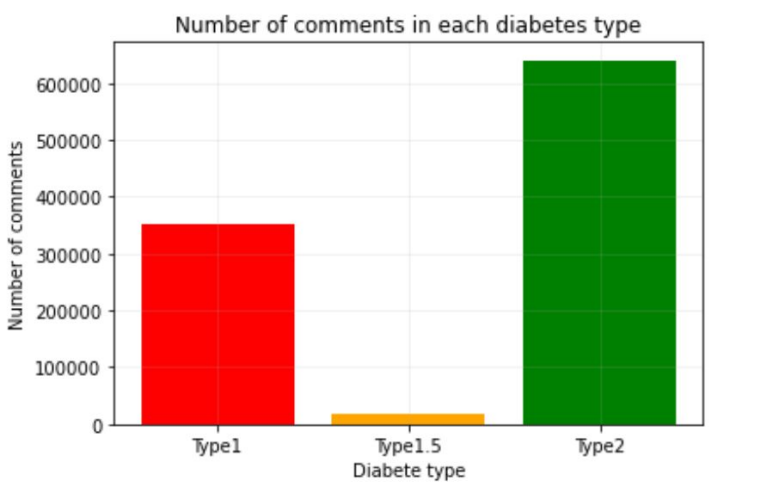
Number of comments extracted

	Diabetes Daiky Forum	Diabetes.co.uk	Total
Type1	128189	222690	350879
Type1.5	9279	9075	18354
Type2	339813	300226	640039

Too few samples in type 1.5 , it is ignored.

Data visuallization





02
Data processing

Spark vs Hadoop

Spark

- APIs available (PySpark/ Scala Spark)
- User-friendly
- In-Memory Computing
- Requires a lot of memory to run
- Not designed for multiple users

Hadoop

- MapReduce (Java/Python)
- Difficult to use
- Reads and writes from disk
- Requires less resources

Spark Environment Setup

- Use Docker container, docker-compose for configuration
- Access via localhost/vscode dev container
- Jupyter Notebook

```yaml
version: "3.7"

services:
  # jupyterlab with pyspark
  pyspark:
    build:
      context: ./
    environment:
      JUPYTER_ENABLE_LAB: "yes"
      NotebookApp.token: ""
      PYTHONPATH: "/usr/local/spark/python/lib/py4j-0.10.9.5-src.zip:/usr/local/spark/python:"
    ports:
      - "8888:8888"
    volumes:
      - ./data:/home/jovyan/work
    command: "start-notebook.sh --NotebookApp.token='' --NotebookApp.password='' --ServerApp.disable_check_xsrf=True"
```

Implementation and result

```
import glob
from pathlib import Path
import shutil
original_csv = glob.glob("./original_csv/*")


shutil.rmtree(Path('./csv'))
Path('./csv').mkdir(parents=True, exist_ok=True)


for file in original_csv:
    csv = pd.read_csv(file, encoding='utf-8')
    print(os.path.basename(file), len(csv.columns), len(csv), csv.columns)
    length=0
    limit=10000
    nth=0
    valid = lambda s: all(i in '0123456789' for i in str(s[csv.columns[0]]))
    while length<len(csv):
        nth+=1
        slice_csv = csv.loc[length:length+limit-1]
        slice_csv = slice_csv[slice_csv.apply(valid, axis=1)]
        slice_csv = slice_csv.dropna(thresh=3)
        slice_csv.to_csv(os.path.join("./csv", os.path.basename(file)[:-4]+'_'+str(nth)+'.csv'), index=False)
        length+=limit
```

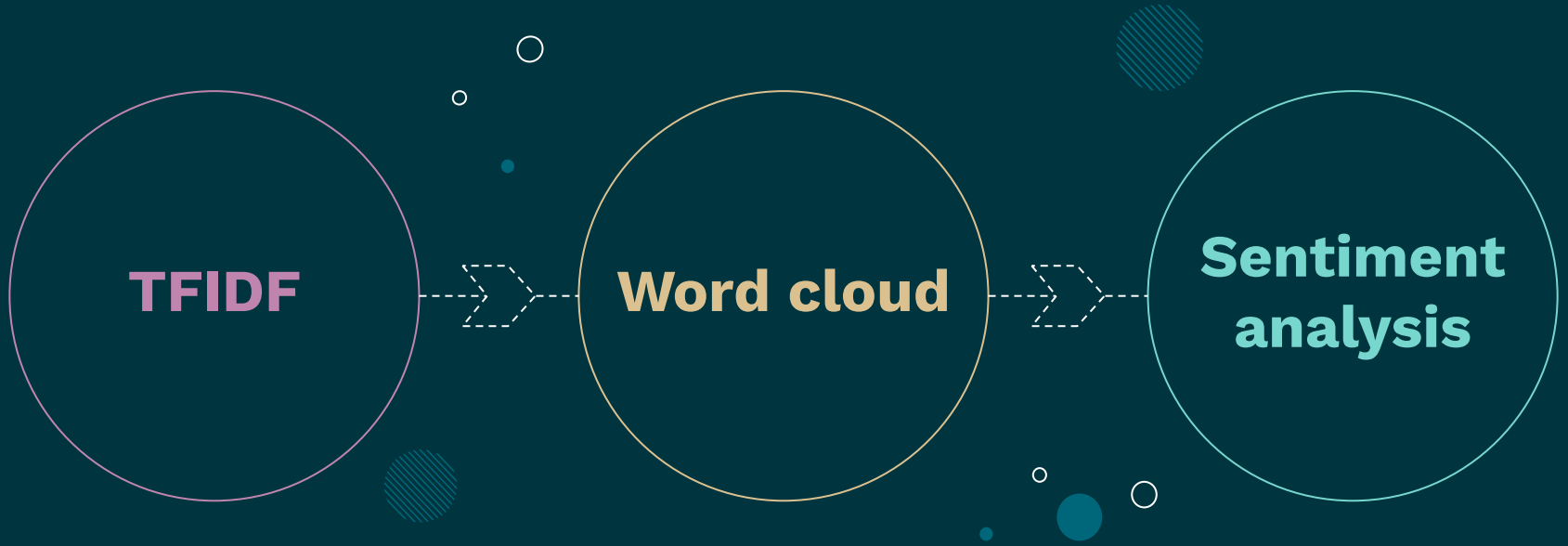
Implementation and result

```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

def transformation(filepath):
    dataset = pd.read_csv(filepath, encoding='utf-8')
    df = spark.createDataFrame(dataset, schema=inputSchema)

    print(os.path.basename(filepath), df.count(), len(dataset))

    punc=r"""[!\"#$%&'()*+,\-.\/:;<=>?@\[\\\]^_`{|}~]"""

    df = df.withColumn("clearText", regexp_replace(
    "text",
    punc,
    " "
    )).withColumn("clearTextNoExtraSpace", regexp_replace(
    "clearText",
    "\\s+",
    " "
    ))

    tokenizer = Tokenizer(inputCol="clearTextNoExtraSpace", outputCol="words")

    tokenized = tokenizer.transform(df)

    remover = StopWordsRemover(inputCol="words", outputCol="filteredWords")

    removed = remover.transform(tokenized)

    wnl = WordNetLemmatizer()

    nltk.download('omw-1.4', quiet=True)
    nltk.download('words', quiet=True)
    nltk.download('wordnet', quiet=True)

    word_set = set(nltk.corpus.words.words())

    @F.udf('array<string>')
    def remove_words(words):
        res=[]
        for word in words:
            new_word=wnl.lemmatize(word)
            if new_word in word_set:
                res.append(new_word)
        return res

    final = removed.withColumn('finalWords', remove_words('filteredWords'))

    drop_list = set(('clearText', 'clearTextNoExtraSpace', 'words', 'filteredWords'))
    final = final.select([column for column in final.columns if column not in drop_list])

    targetPath = os.path.join(output_folder, "output_"+os.path.basename(filepath))
    final.toPandas().to_csv(targetPath, index=False)

    print('Done')
    return targetPath
```

topicNumber	topic	text	finalWords
0	Managing exercise an…	A number of members have posted recently about sport and exercise …	['achievement', 'personally', 'regularly', 'challenge', 'rece…
0	Managing exercise an…	I train and compete in powerlifting, which is a strength sport based aro…	['hypertrophy', 'hypertrophy', 'competition', 'competition', …
0	Managing exercise an…	I used to play tennis, but had some issues with my coach also didnt ha…	['everywhere', 'extremely', 'recommend', 'anyways', 'bicy…
0	Managing exercise an…	I never called myself a runner when I could just run whenever I wanted…	['preparation', 'immediately', 'commitment', 'arithmetic', 'a…
0	Managing exercise an…	Hi @NoKindOfSusie I would get into see your GP and just get checke…	['checked', 'recover', 'symptom', 'forward', 'longer', 'really'…
0	Managing exercise an…	I'm checking my levels lots and lots, definitely before I go running and …	['circumstance', 'combination', 'definitely', 'apparently', 'p…
0	Managing exercise an…	Hi @Juicyj . . . I recently read the book "Bright Spots and Landmines"…	['appropriate', 'interesting', 'incorporate', 'adjustment', 'aff…
0	Managing exercise an…	Hi @NoKindOfSusie - Do you have any quick acting insulin on board …	['absorption', 'something', 'diagnosis', 'remember', 'insuli…
0	Managing exercise an…	I've only been diagnosed for about 8 weeks now and have found exerc…	['wheelbarrow', 'something', 'otherwise', 'sensitive', 'brea…
0	Managing exercise an…	Circus that's basically my experience, any kind of serious exercise just…	['experience', 'basically', 'treatment', 'breakfast', 'vegetab…
0	Managing exercise an…	Oh bless you @Juicyj - keen but not doing well with it right now! I can …	['considering', 'afterwards', 'experiment', 'preferably', 'co…
0	Managing exercise an…	For what it's worth (which is nothing) I am not that keen on the whole p…	['dangerously', 'personally', 'unmodified', 'personally', 'inc…
0	Managing exercise an…	Hi @NoKindOfSusie I felt exactly the same before I got my pump, I wa…	['flexibility', 'confident', 'attached', 'actually', 'exercise', 'e…
0	Managing exercise an…	I am one of the people who has struggled and asked about diabetes a…	['replacement', 'achievement', 'treadmill', 'treadmill', 'som…
0	Managing exercise an…	This is music to my ears. I'm hoping a libre will help me along. My DS…	['attempt', 'session', 'enable', 'music', 'along', 'least', 'mon…
0	Managing exercise an…	Hey @lely I started in the couch to 5k, really worthwhile doing this pro…	['confidence', 'management', 'building', 'program', 'insulin…
0	Managing exercise an…	Also to add that I do parkruns too @lely if you start to flag there? alw…	['encouragement', 'satisfaction', 'friendly', 'feeling', 'morni…
0	Managing exercise an…	I did attend my first Park run last sat as an extra go for the couch to 5k…	['appointment', 'completely', 'different', 'tomorrow', 'daug…
0	Managing exercise an…	I've just come back from an amazing diabetes and exercise weekend i…	['professional', 'opportunity', 'opportunity', 'tribulation', 'inf…
0	Managing exercise an…	Hi @katmcd your weekend sounds great was Roddy Riddle a famous t…	['marathon', 'speaking', 'weekend', 'riddle', 'famous', 'run…
0	Managing exercise an…	I could have written your post today! I seem to spend most of my life …	['appointment', 'unpleasant', 'consultant', 'diagnosis', 'oth…
0	Managing exercise an…	Wow, I did this 18 months ago and it was a FABULOUS weekend (apa…	['confidence', 'challenge', 'wonderful', 'recommend', 'fabu…
0	Managing exercise an…	Roddy Riddle was there. His before dinner presentation was phenome…	['presentation', 'information', 'necessarily', 'importantly', '…
0	Managing exercise an…	It was wonderful. In a way I feel so privileged to have been able to go …	['information', 'information', 'immediately', 'privileged', 'ex…
0	Managing exercise an…	Fabulous post @katmcd It is definitely about being your own detectiv…	['definitely', 'capability', 'detective', 'breathing', 'finishing', …
0	Managing exercise an…	So when I read those slides above I realise why I could ride my horse …	['consistently', 'approaching', 'struggling', 'absolutely', 'ad…
0	Managing exercise an…	Hi @Circuspony - I have found that I need carbs before I exercise with…	['difference', 'afterwards', 'throughout', 'breathing', 'breat…
0	Managing exercise an…	Interesting to hear peoples experiences with running... I'm currently tra…	['interesting', 'interesting', 'experience', 'struggling', 'curre…
0	Managing exercise an…	Mathematics time. I have no idea if this is right, any ideas gratefully re…	['mathematics', 'kilocalorie', 'gratefully', 'reasonably', 'the…
0	Managing exercise an…	Hi @NoKindOfSusie - It depends what you mean by high ? What level…	['supplementation', 'approximately', 'carbohydrate', 'unac…
0	Managing exercise an…	All credit to Ian Gallan,Alistair Lumb and James Moran for the slides (fr…	['according', 'direction', 'splitting', 'diabetes', 'treating', 'in…
0	Managing exercise an…	I cycle reguraly to college. So this is bassically what i do: Wake up at 7…	['protective', 'breakfast', 'confident', 'equipment', 'climbin…
0	Managing exercise an…	Hi, I'm new here, I've had T1 since the early 80s, I managed to play a…	['prescription', 'management', 'sometimes', 'weakening', '…
0	Managing exercise an…	Hi @kev-w Thanks for posting your update on your exercise, I proceed…	['temperature', 'beforehand', 'afterwards', 'sometimes', 'n…
0	Managing exercise an…	Thanks for your reply, I'm similar to you with hypo awareness hence …	['awareness', 'recommend', 'diabetes', 'hospital', 'standar…
0	Managing exercise an…	Hi @kev-w Personally I would try and stay below 9mmol/l for this simp…	['personally', 'sometimes', 'providing', 'brilliant', 'exercise'…
0	Managing exercise an…	Hi JuicyJ and thanks for your reply, A long time ago I needed an ambu…	['unconscious', 'confidence', 'roundabout', 'ambulance', '…
0	Managing exercise an…	Hi @kev-w Crumbs I can understand how you feel, I was swimming wi…	['understand', 'incredibly', 'experience', 'swimming', 'dau…
0	Managing exercise an…	New longest run today (8 miles) in training for my half! But completely …	['investigation', 'completely', 'experience', 'experience', 'a…

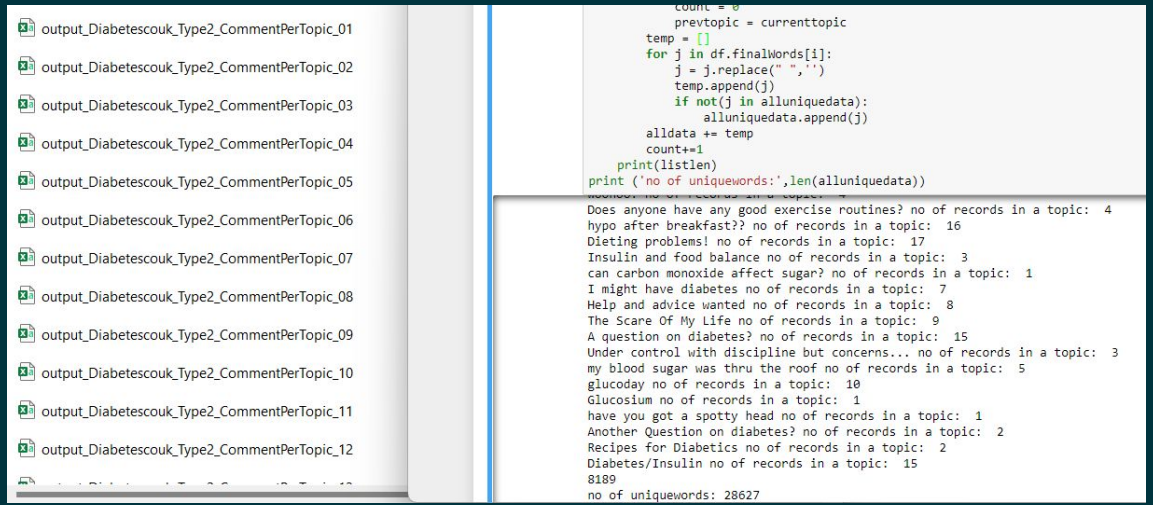
03
Methods & Results

Analysis methods



TFIDF – Implementation

Step 1: Separate and combine the information
- **Aims: Obtaining unique words, group tokenized words of comments by topic**



Number of comment records:
Type 1: 130,000 Type 2: 600,000

Number of unique words:
Type 1: 28627 Type 2: 31327

TFIDF – Implementation

26	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
27	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
28	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
29	0.000000	0.00000	0.000000	0.064462	0.000000	0.000000	0.000000	0.000000	0.045160	0.000000	...	0.0000
30	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
31	0.062738	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
32	0.000000	0.00000	0.038945	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
33	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
34	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.034461	0.000000	...	0.0000
35	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
36	0.000000	0.00000	0.000000	0.098420	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
37	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.035660	0.000000	0.000000	...	0.0000
38	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
39	0.000000	0.00000	0.000000	0.015532	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
40	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
41	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
42	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
43	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
44	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
45	0.000000	0.00000	0.000000	0.034306	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
46	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
47	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000
48	0.000000	0.00000	0.000000	0.039356	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.0000

49 rows × 3806 columns

Step 2

- Reduce the term frequency that seldom appears
- Filter out those terms from topic documents

Step 3

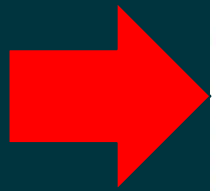
- Convert tokens back to a list for each topic

```
temp = []
tempcommentlist = eval(df.finalWords[i])
for j in tempcommentlist:
    j = j.replace(" ",'')
    temp.append(j)
    if not(j in alluniquedata):
        alluniquedata.append(j)
alldata.append(temp)
```

TFIDF – Implementation

```
print(databytopic[1:7])
```

```
are'], ['follow', 'military', 'diet', 'treat', 'week', 'pizza', 'hut'], ['another', 'fantastic',
ook', 'insulin', 'walsh', 'ruth', 'first', 'went', 'pump', 'yr', 'ago', 'recommend', 'sugar', 'als
o'], ['many'], ['hello', 'guy', 'diabetic', 'type', 'well', 'working', 'new', 'channel', 'help',
nspire', 'diabetic', 'link', 'one', 'video', 'find', 'useful', 'please', 'give', 'like', 'share',
'many', 'thanks'], ['best', 'book', 'ever', 'like', 'pancreas', 'sugar', 'waste', 'time'], ['grea
t', 'recommendation', 'really', 'fine', 'tuned', 'control', 'resource', 'think', 'like', 'pancrea
s', 'must', 'read', 'make', 'sure', 'get', 'recent', 'edition', 'active', 'person', 'really', 'enj
y', 'diabetes', 'type', 'movement', 'great', 'listen', 'drive', 'walking', 'also', 'follow', 'soci
l', 'medium', 'post', 'regularly', 'resource', 'able', 'bring', 'time', 'range', 'average', 'day',
'auto', 'mode'], ['sensor', 'monitor', 'diabetes', 'powerful', 'tool', 'let', 'glucose', 'reading'
'alarm', 'without', 'scanning']], [['come', 'say', 'hi'], ['hi', 'tested', 'moment', 'join', 'get'
'positive', 'test', 'back'], ['hi', 'flora', 'look', 'forward', 'hearing', 'result', 'm'], ['sen
t'], ['six', 'family', 'vibrant', 'community', 'member', 'search', 'love', 'see', 'well'], ['hi',
'five', 'year', 'ago', 'without', 'always', 'happy', 'share', 'experience'], ['hi', 'm', 'rare',
ype', 'apparently', 'almost', 'year', 'ago', 'mum', 'also', 'found', 'brother', 'carry', 'gene',
um', 'told', 'type', 'insulin', 'year', 'brother', 'daughter', 'born', 'due', 'gene', 'lot', 'goin
g', 'u', 'nice', 'meet'], ['hi', 'nice', 'meet', 'haven', 'checked', 'm', 'mum', 'sister', 'm',
sulin', 'always', 'thought', 'type', 'year', 'insulin', 'want', 'change', 'tablet', 'mum', 'siste
r', 'treat', 'tablet']]]
```



```
t way morning right back change food sure lot good month test morning morning night eating morning
morning morning last day last night low last day much last night morning right one hour getting ba
k start also eating still make even good morning morning last night morning morning morning going
ry good like long time eating morning morning even bit getting blood sugar one time back hour also
go back really thanks hi hope going well like blood glucose take get back best good good high hi k
ow try really low day like start day day day food need make eat also food think day get think need
day m eating good u know high low morning think last month last test bit last morning though night
lot night even work well back go morning re still got new hour going right way good know food foru
eat meal see hope re well morning morning morning morning morning still well high last day still t
anks forum day morning morning day re time right hope help morning low though really good day morn
ng last night morning good morning last night morning thing morning morning morning morning sugar
evel going sugar m eat lot day sugar morning first one night got morning hi hope never really hope
see thanks bit since know bit better morning morning morning diabetic better morning time day well
need insulin morning thing year week hope better much morning morning morning day forum morning bi
morning morning also morning morning go bit diabetes blood glucose problem morning good morning la
t night last night morning morning morning getting around last night morning morning first test mo
ning week blood sugar level blood test much see sugar life hour one around got morning day diabete
forum morning keep always something time keep well morning morning back well level last night m be
t good get try control even diabetic long time day want morning last week good morning morning m e
```

Step 4:
- count term frequency,
- sort using Dict.Key

```
['insulin', 132716]
['get', 103014]
['time', 95565]
['day', 88989]
['m', 82134]
['like', 81856]
['one', 80312]
['diabetes', 79813]
['know', 71768]
['good', 68579]
['year', 67039]
['think', 62990]
['need', 61414]
['go', 58333]
['type', 58101]
['thing', 57619]
['take', 57521]
['low', 57247]
['much', 55783]
```

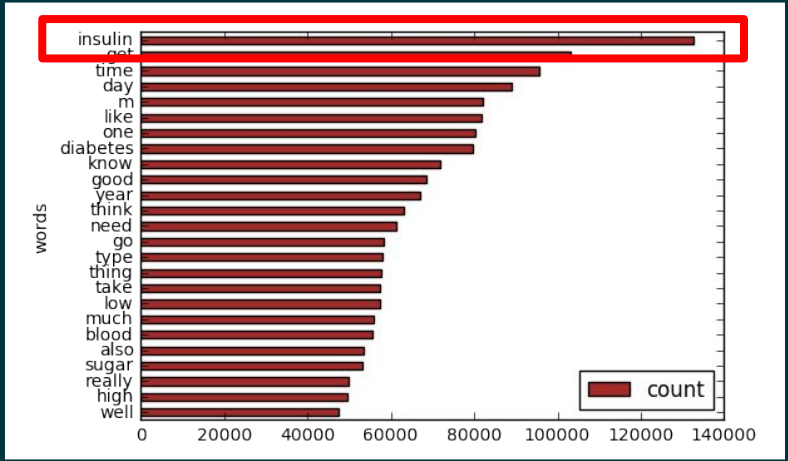
```
wordnum = dict.fromkeys(alluniquedata,0)
for j in databytopic:
    for i in j:
        wordnum[i]+=1
for i in wordnum:
    tmp = i.replace("'",'')
    count.append([tmp,wordnum[i]])

count.sort(key = lambda x: x[1],reverse=True)
for i in range(100):
    print(count[i])
    w.append(i)
    c.append(count[i][0])
```

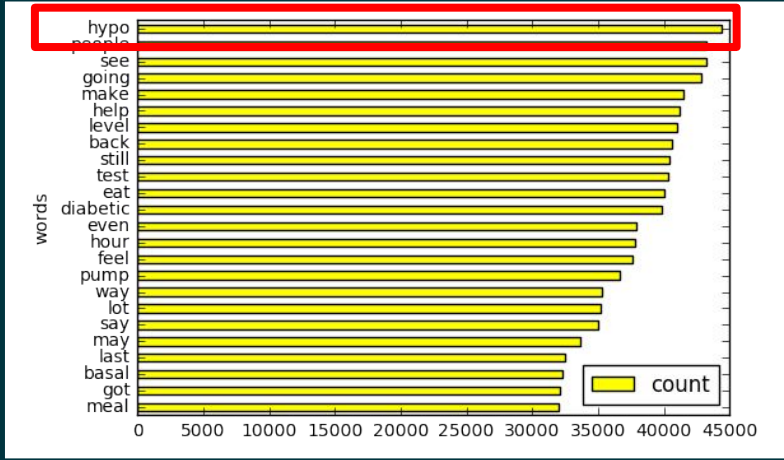
TFIDF - Vectorizer

Step 5:
- Vectorizing, comparing the relationships and sum up all the data.

insulin	3540.164712
get	2606.212233
diabetes	2460.584529
time	2370.000879
day	2288.745733
one	2108.582754
like	2105.233978
year	2011.054335
know	1959.132254
type	1878.675212
good	1868.474658
blood	1796.046930
sugar	1784.174909
low	1735.973984
take	1717.246669
need	1706.706016
think	1687.631655
go	1647.320302
high	1602.721953
thing	1592.951125
hypo	1560.423800
also	1558.534941
much	1545.219214
really	1467.470916
level	1456.223274
pump	1430.845300
well	1428.461123
help	1425.771243
work	1420.677228
test	1417.731543

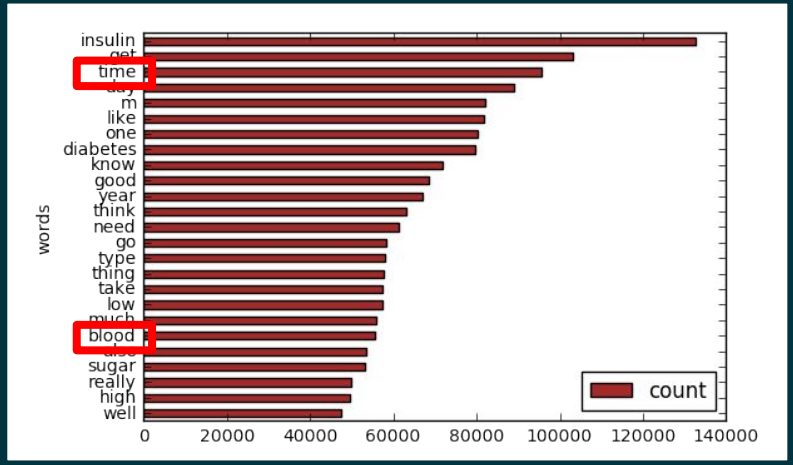
TFIDF – Type 1 Data visualization

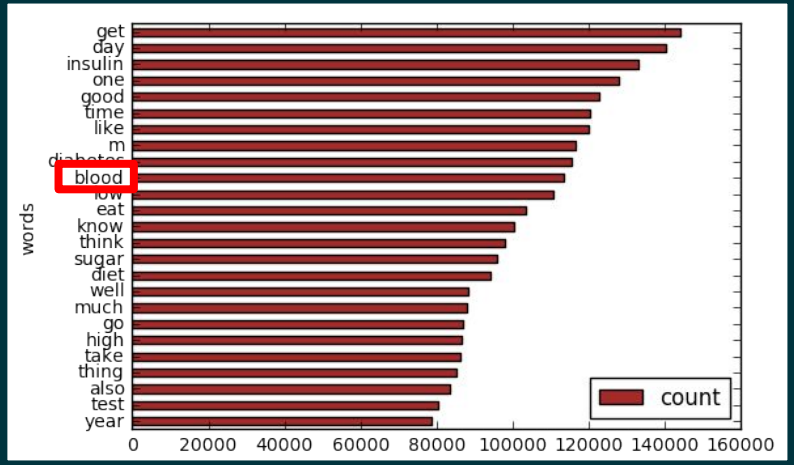


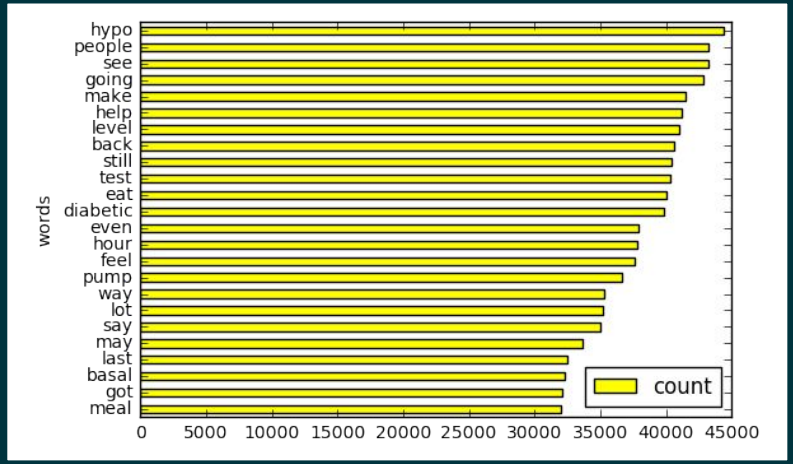


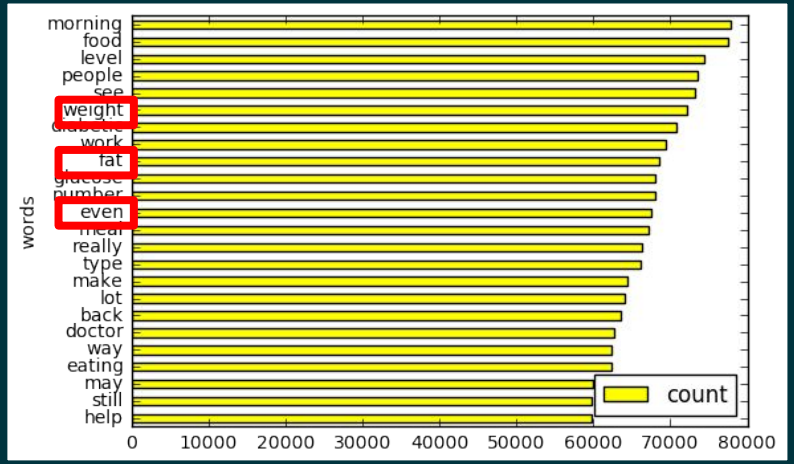
28602	reclassify	1
28603	zoning	1
28604	emote	1
28605	grumbler	1
28606	belfry	1
28607	urinator	1
28608	egocentric	1
28609	embellishment	1
28610	precognition	1
28611	overexposure	1
28612	citizenry	1
28613	grama	1
28614	ornamental	1
28615	culpable	1
28616	gawk	1
28617	missive	1
28618	mildew	1
28619	forgivable	1
28620	milepost	1
28621	retroaction	1
28622	sacrilege	1
28623	coloration	1
28624	sally	1
28625	lugged	1
28626	sneering	1

TFIDF – Type 1 & 2 Data visualization









TFIDF - Comparison

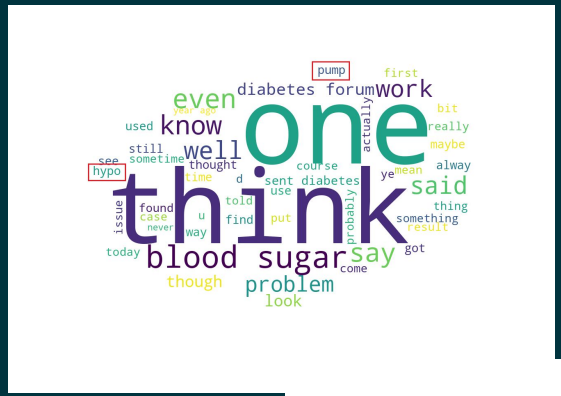
Type 1

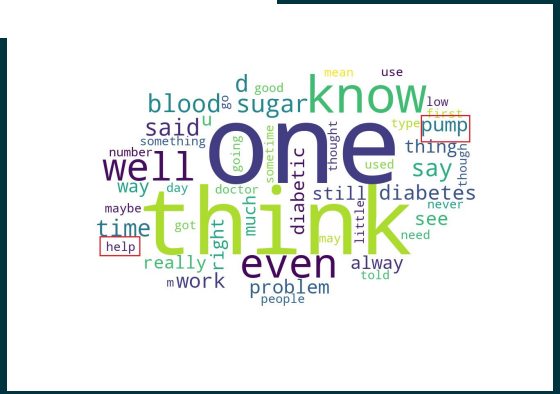
insulin	3540.164712	long	1026.767680
get	2606.212233	never	1025.900250
diabetes	2460.584529	find	1020.338705
time	2370.000879	food	1017.646097
day	2288.745733	morning	1015.332016
one	2108.582754	many	1010.276817
like	2105.233978	re	1006.311510
year	2011.054335	always	992.992570
know	1959.132254	said	988.615930
type	1878.675212	glucose	984.298058
good	1868.474658	bolus	982.599766
blood	1796.046930	first	968.552412
sugar	1784.174909	bit	960.074778
low	1735.973984	hope	949.167428
take	1717.246669	something	946.812257
need	1706.706016	right	929.181103
think	1687.631655	life	897.064597
go	1647.320302	around	888.812052
high	1602.721953	since	879.734660
thing	1592.951125	give	876.527389
hypo	1560.423800	change	873.871809
also	1558.534941	eating	872.212981
much	1545.219214	getting	871.453068
really	1467.470916	might	862.505326
level	1456.223274	start	851.755528
pump	1430.845300	best	846.254900
well	1428.461123	little	840.584212
help	1425.771243	though	831.421372
work	1420.677228	different	813.803464
test	1417.731543	every	810.875488

Type 2

get	3267.326107	try	1311.958841
day	3199.279641	never	1278.506970
insulin	3174.696744	sure	1276.043596
diabetes	2994.254220	something	1234.425613
blood	2927.652504	forum	1230.926696
one	2895.595493	right	1202.961058
good	2890.919966	long	1200.846027
low	2856.790278	new	1174.691767
time	2795.990896	might	1162.730993
like	2735.508029	re	1134.296770
sugar	2683.815749	bit	1132.662600
eat	2610.247171	always	1120.032844
test	2499.970389	give	1116.183615
know	2468.399713	little	1112.856433
take	2421.717131	around	1094.538708
well	2340.801581	different	1074.563955
think	2268.106423	hope	1074.505882
high	2235.549359	getting	1067.261559
go	2216.958829	though	1062.068395
level	2192.194099	change	1060.054187
year	2191.663919	night	1030.804933
also	2172.405416	best	1013.388952
thing	2087.422616	every	985.897642
much	2072.548900	start	924.300507
doctor	2061.185484	life	887.890524
food	2044.152726	hypo	738.594607
meal	1994.747560	unit	664.419089
diabetic	1988.859049	basal	287.155977
need	1963.047417	pump	233.388939
work	1950.729093	bolus	201.639337

Word cloud





Type 1:

- Common and frequent words are **generic**
- Blood sugar, pump (device that release insulin) can be their concerns
- Neutral sentiments, but tend to need more assistance
  e.g. help

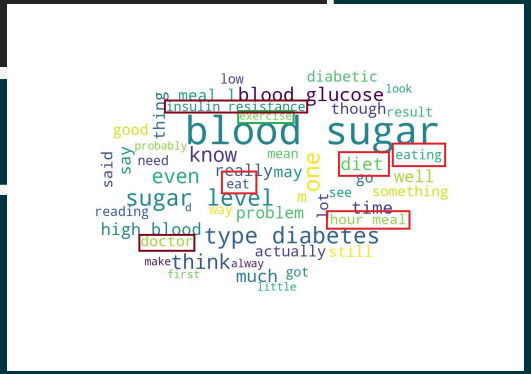
Type 2:

- Common and frequent words: *blood sugar*, *think*, *one*
- Some useful terms for investigation can be noticed *e.g. blood sugar*
- Relatively more positive sentiments *e.g. well done, believe, right*

```
#type2 concat
type2_dfs=couk_type2_df.append(daily_type2_df)
type2_dfs

#blood sugar word cloud
bs_filter = type2_dfs['text'].str.contains("blood sugar", na=False)

type2_bs=[]
for row in type2_dfs[bs_filter]['finalWords']:
    data=eval(row)
    type2_bs.extend(data)


wordcloud_bs=WordCloud(width=1440, height=900, max_words=50, background_color="white"
, mask=mask).generate(' '.join(type2_bs))

plt.imshow(wordcloud_bs, interpolation='bilinear')
plt.axis("off")
plt.figure( figsize=(2,1) )
plt.show()
```

Using the term **'blood sugar'** to filter the data

Generate wordcloud



What possible factors causes blood sugar as a frequent term in type 2?

- eating habits (meals, eating keywords)
- insulin resistance

What they need to do?

- transform their diet habit
- exercise (being active makes your body more sensitive to insulin)
- seek for doctor

Sentiment analysis

- Sentiment analysis tool: **VADER** (Example)

```
import pandas as pd

#vader's download
import nltk
nltk.download('vader_lexicon')

from nltk.sentiment.vader import SentimentIntensityAnalyzer

sentiment=SentimentIntensityAnalyzer() #initialize

#Examples
list_=["This phone is extremely bad.","This phone is not that good.","This phone is very good."]
sentiment_list=[]

print("Example 1 Only using vader\n")
for i in list_:
     print(i,': ',sentiment.polarity_scores(i))
print("\n--------------------------------------------------\n")

print("Example 2 Only using vader 's compound (Sentiment score)\n")
for i in list_:
     print(i,': ',sentiment.polarity_scores(i)['compound'])
print("\n--------------------------------------------------\n")

print("Example 3 Only using vader 's compound (Sentiment score) and append\n")
for i in list_:
     sentiment_list.append(sentiment.polarity_scores(i)['compound'])

for i in range(len(list_)):
     print(i,': ',sentiment_list[i])
print("\n--------------------------------------------------\n")
```

Code

Output

```
Example 1 Only using vader

This phone is extremely bad. :  {'neg': 0.487, 'neu': 0.513, 'pos': 0.0, 'compound': -0.5849}
This phone is not that good. :  {'neg': 0.325, 'neu': 0.675, 'pos': 0.0, 'compound': -0.3412}
This phone is very good. :  {'neg': 0.0, 'neu': 0.556, 'pos': 0.444, 'compound': 0.4927}

--------------------------------------------------


Example 2 Only using vader 's compound (Sentiment score)

This phone is extremely bad. :  -0.5849
This phone is not that good. :  -0.3412
This phone is very good. :  0.4927

--------------------------------------------------


Example 3 Only using vader 's compound (Sentiment score) and append

0 :  -0.5849
1 :  -0.3412
2 :  0.4927

--------------------------------------------------
```

Sentiment analysis

- In dataframe, column can be a list of data -> for loop
- Drop na is important
- Order is needed , so for loop is slow but stable

In [16]:
```
def add_sentiment(j):
    t1["sentiment"][j]=sentiment.polarity_scores(t1["text"][j])['compound']
```

In [17]:
```
#calculating time
import time
```

In [25]:
```
#normal computing
start=time.time()
for i in list(range(1,1001)):
    add_sentiment(i)
end=time.time()
normal_exe_time=end-start
print("The time of normal computing is:",normal_exe_time," s")
```

```
C:\Users\kiusandy\AppData\Local\Temp\ipykernel_33456\3648476584.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  t1["sentiment"][j]=sentiment.polarity_scores(t1["text"][j])['compound']
```

```
The time of normal computing is: 13.280495405197144  s
```

Sentiment analysis

Out[28]:

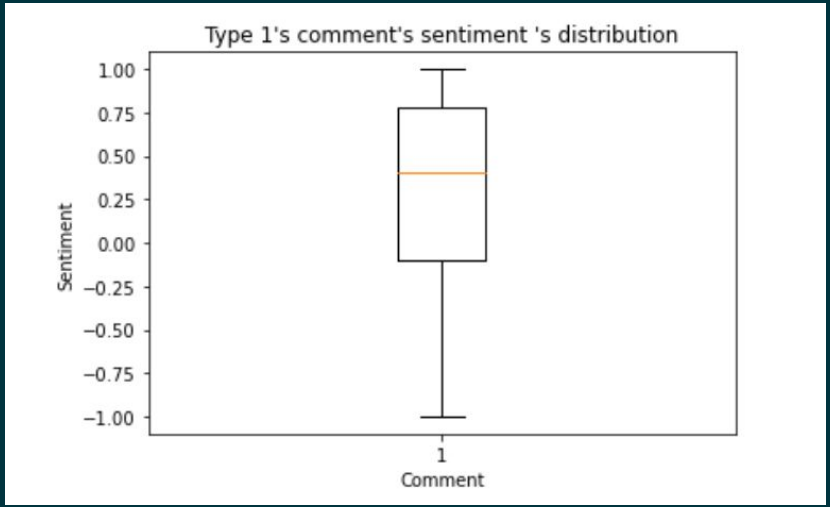
	topic	text	sentiment
1	Learning Center - Type 1 Diabetes	For up to date information about Type 1 Diabet...	0.5661
2	Learning Center - Type 1 Diabetes	I have a question about insulin antibody testi...	-0.7096
3	Learning Center - Type 1 Diabetes	In general, a reference range for any given te...	0.6335
4	Learning Center - Type 1 Diabetes	Thank you for the reply. I think you must be ...	-0.5267
5	Learning Center - Type 1 Diabetes	Medtronics insulin pump with MiniMed Quick -se...	-0.7644
...	...	...	...
348714	Test strips	I test regularly, about 4 times a day, so in a...	0.0000
348715	Test strips	I test on adverage just over 50 strips a week,...	-0.9775
348716	Test strips	Well now you say it, yes and not very prett !...	0.6114
348717	Welcome to the Type 1 Diabetes forum	Hope this is useful to our visitors...\n\nDan	0.7003
348718	Welcome to the Type 1 Diabetes forum	Random new forum outta nowhere! Let's have a...	0.0000

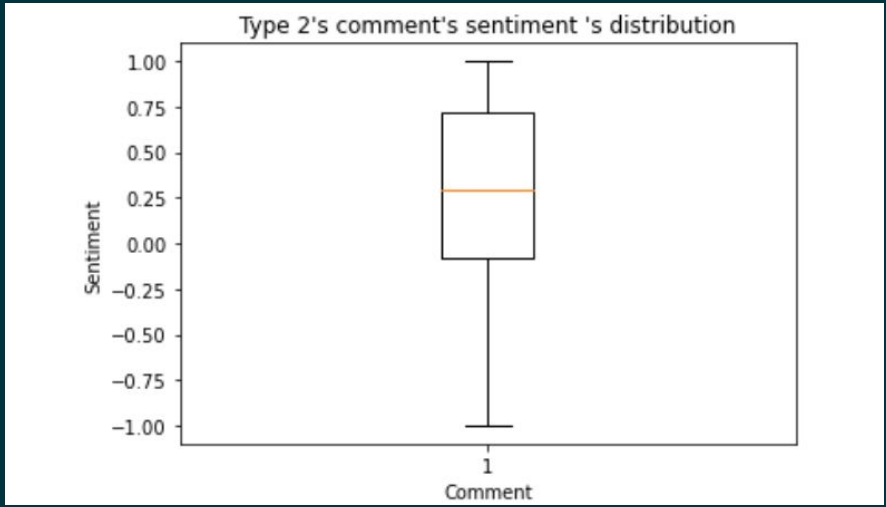
348718 rows × 3 columns

Sentiment analysis

Diabetes type	Time
Type 1	7587.37 s
Type 2	9154.32 s

Sentiment analysis





Statistics of sentiments
Maximum: 0.9998
Upper quantile: 0.7783
Mean: 0.2658757233638426
Lower quantile -0.1025
Minimum: -0.9997

Statistics of sentiments
Maximum: 0.9998
Upper quantile: 0.71715
Mean: 0.22697844072806617
Lower quantile -0.07515
Minimum: -0.9996

Sentiment analysis result -type 1

	Topic	Sentiment
1	"More Ways to Cope With Type 1 Diabetes"	-0.98710
2	the revolution starts now	-0.98480
3	Rapid D sets-just be aware of this	-0.98060
4	Addison's * Diabetes Questions?	-0.97890
5	Help With Finding Alternate Diabetes Supplier	-0.97880
6	LADA	-0.97120
7	how does stress affect your bg?	-0.97060
8	you will just have another honeymoon perio d before your immune system destroys that pancreas too	-0.96360
9	Neuropathic Pain	-0.96330
10	Been There, Done.....it all??	-0.96330
11	Kidney/Stomach/Gland Issues	-0.95650
12	Gastroparesis bolusing	-0.95450
13	Floaters after Haemorrhage	-0.95240
14	emotionally drained	-0.94410
15	Now I want a suggestion	-0.94045
16	What's so special about " Serums "_?	-0.93860
17	GP telling me nothing is wrong, leading to bad infection	-0.92270
18	Is it because i had a hypo????? anyone hel p??	-0.92240
19	Type 1 and glandular fever	-0.91910
20	is Apologies for rant!	-0.91905

Sentiment analysis result -type 1

(Obtained by comments)

- Side effect of hypo : **irretional anger**
  -> need **more people's care and understand**

- Type 1 patients need injection of insulin, meter -> **high living cost**
- Many expensive holiday activities **cannot be afforded**
- lower life quality

->need more support from their relatives , friends, even society

Sentiment analysis result -type 2

topic	Sentiment score
How dos one cope with fake lows	-0.91985
Little bit goes a long way.	-0.92870
Finally able to check bg after regular Saturday dinner - surprised	-0.92910
Diabetics face risk on drug choices	-0.93490
Diabetes and Gallstones	-0.93530
The Predators on the Prowl of our Beta Cells (hunting license not required)	-0.94290
feeling generally unwell	-0.94345
Anxiety, Neuropathy, Back Pain, Unsteady BG Readings @ Hunger	-0.94580
Annual checkups	-0.95430
Trulicity Advise	-0.95640
Getting back on track again after my fall injury	-0.96400
Gastric Flu	-0.96450
Feeling Drained Again	-0.96490
Drink up girls	-0.97070
DAPAGLIFLOZIN/METFORMIN	-0.97140
Bad nights make me feel deathly	-0.97540
high bg	-0.97840
Ever have hours or days where you just don't care?	-0.98430
Breathing troubles	-0.99260
Alcohol and hypo..hypo and depression	-0.99560

Name: sentiment, dtype: float64

Sentiment analysis result -type 2

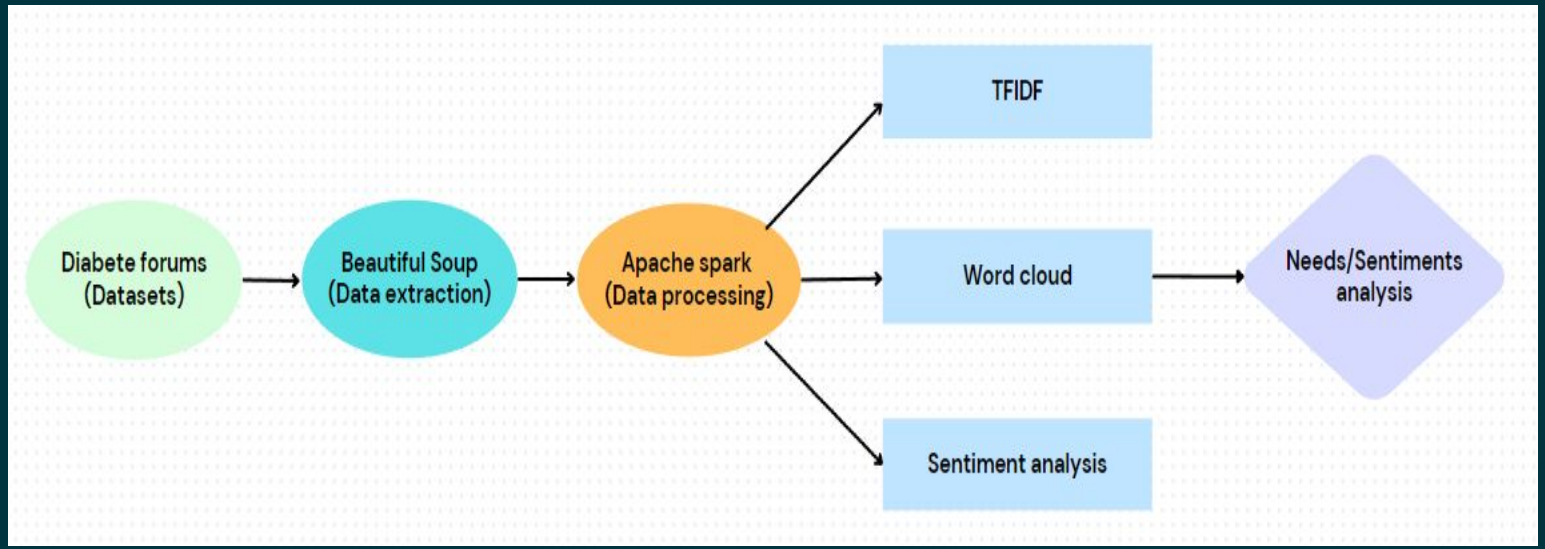
(Obtained by comments)

- Need to **lose weight quickly** -> **easily be stressed**

- The drug, **Avandia** (a drug for diabete type 2 adults) is still used.
  Use : lowers blood sugar
  Side effect : **30% higher risk of a heart attack**
  -> A **worrying issue** in their treatment.

04
Conclusion

Conclusion


Diabete forums
(Datasets)
Beautiful Soup
(Data extraction)
Apache spark
(Data processing)
TFIDF
Word cloud
Sentiment analysis
Needs/Sentiments
analysis

Type1 diabetes: 	**Type2 diabetes:** 
• major sentiments are relatively **neutral to negative** • **insulin and hypo** are their major concerns • need **support from friends and family members**	• major sentiments are relatively **neutral to positive** • **weight and diet** are their major concerns • needs to be **self-motivated**

The End