Machine Translation

Clause Restructuring for Statistical Machine Translation

Summary

Aqeel Labash

Task 1

a) Merging to one dataset

done that using this code (python):

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
dfa = pd.read_csv('pupila.csv',sep='\t')
dfc = pd.read_csv('pupilc.csv',sep='\t')
dfd = pd.read_csv('pupild.csv',sep='\t')
dfa['C'] = dfc['C']
dfa['D'] = dfd['D']
dfa.to_csv('ACD.csv',index=False)
```

b)Get Some heatmap

```
ax = plt.subplot(1,1,1)
ax.pcolor(dfa[dfa.columns[[1,2,3]]].as_matrix())
ax.set_xticklabels(['','A','','C','','D',''])
plt.show()
```

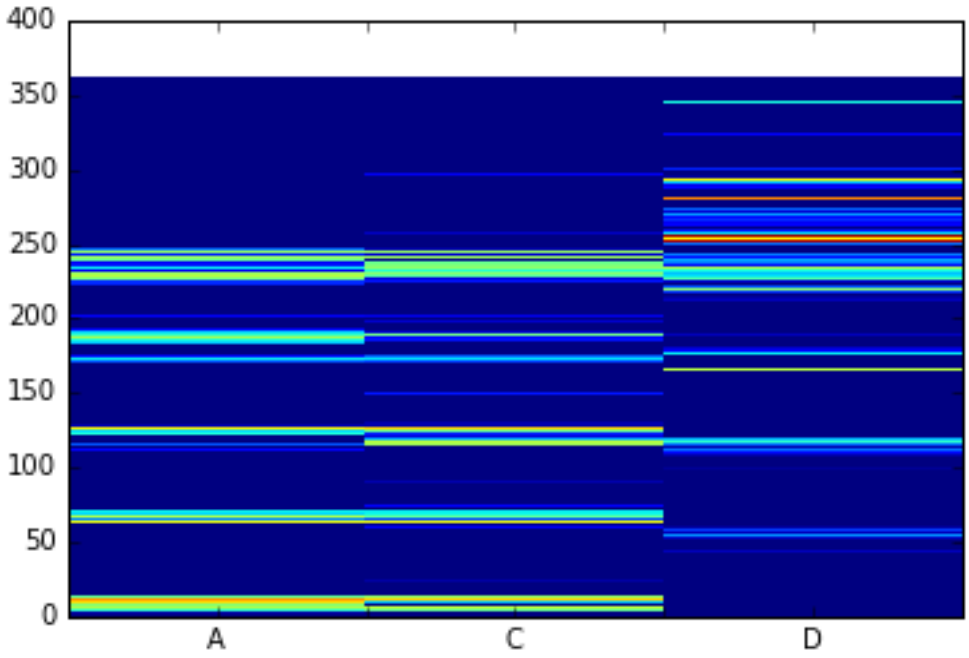

Figure 1: Heat map of students

c)When Lessons Start

For this actually I depended on the fact that usually during lessons people more focused on one thing so the activity of the student pupil should be much less.
So for the start,end times I would say :

Class #	School 1	School 2
1	8:15→9:00	8:00→8:45
2	9:15→10:00[1]	9:00→9:45
3	10:15→11:00[2]	10:00→10:45
4	11:10→11:55[3]	11:00→11:45 [4]
5	No movement on students `A,C`	12:00→12:45[5]
6	No movement on students `A,C`	13:00→13:45

[1]Looks like there were some activity at the end of this class, activity started around 9:55 so I believe the class ended

1) Most active student

```
dfa['A'].sum(),dfa['C'].sum(),dfa['D'].sum()
```

The previous code give :A:5044, C:4663, D:4815) we can notice that Student A was the most active student

2) Which two students from same school

Students A,C were from the same class. Simply if we compare the active time we will find that students A,C values are highly correlated.

```
dfa.corr()
```

	A	C	D
A	1.000000	0.802555	0.052867
C	0.802555	1.000000	0.095510
D	0.052867	0.095510	1.000000

Table 1: Correlation between students pupils

From Table 1 we can see how much students A,C are correlated.

3)

The breakfast for students A,C was from 8:05 → 8:15

4) Which student has the physical class

Student D has the physical class. This student has high pupil activity for long time.Fig. 2 show this activity in bottom left plot. Which make sense, since physical class student has to move his eye much more than traditional class.

```
plt.figure(figsize=(16,10))
ax = plt.subplot(2,2,1)
ax.set_title('Student A')
ax.set_xlabel('Minutes')
ax.set_ylabel('pupil activity')
ax.bar(dfa.index,dfa.A)
ax = plt.subplot(2,2,2)
ax.set_xlabel('Minutes')
ax.set_ylabel('pupil activity')
ax.bar(dfa.index,dfa.C)
ax.set_title('Student C')
ax = plt.subplot(2,2,3)
ax.set_xlabel('Minutes')
ax.set_ylabel('pupil activity')
ax.bar(dfa.index,dfa.D)
ax.set_title('Student D')
```

before the schedual

[2]Seems starange time to start and end but make sense depending on the data. For accurate the class ended at 10:52

[3]I selected this end depending on the class duration. but at the end of the class high activated does exist around 11:45

[4]This class contained some activity

[5]contained a lot of regular activate (I think this one is the sport class

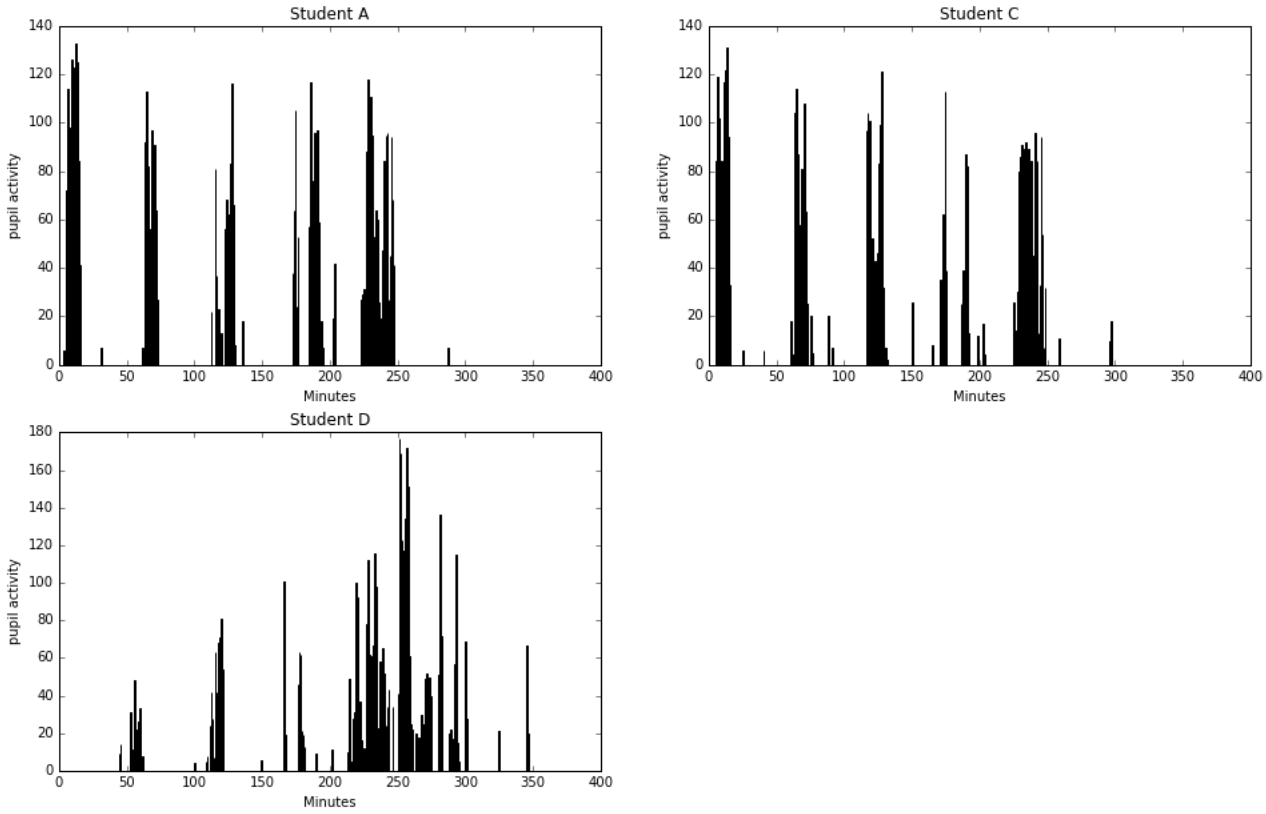

Figure 2: bar plot for students activity

5) Difficulties with 100 students.

Although we can see how things going with the current dataset, we don't know what actually happen. We might think that this less activity might be class but it might be just sleeping student :). On the large datasets we will have to deal with the noise of this kind all the way. The problem will be much harder with students from different classes and different schools (assuming untagged samples). I think one of the most problems will be detecting hardware deficit.

Task 2

1)Most Expensive Bills

```
df.sort(columns=['cost'],ascending=False).head()
```

Table 2 show the most 3 expensive codes

Code	Cost
I22	11413.0
I21	6657.6
F20	4530.5

Table 2: Most expensive codes

1. **I22 Subsequent myocardial infarction**
    - I22.0 Subsequent myocardial infarction of anterior wall
    - I22.1 Subsequent myocardial infarction of inferior wall
    - I22.8 Subsequent myocardial infarction of other sites
    - I22.9 Subsequent myocardial infarction of unspecified site
2. **I21 Acute myocardial infarction**
    - I21.0 Acute transmural myocardial infarction of anterior wall
    - I21.1 Acute transmural myocardial infarction of inferior wall
    - I21.2 Acute transmural myocardial infarction of other sites
    - I21.3 Acute transmural myocardial infarction of unspecified site
    - I21.4 Acute subendocardial myocardial infarction
    - I21.9 Acute myocardial infarction, unspecified
3. **F20.0 Schizophrenia**
    - F20.0 Paranoid schizophrenia
    - F20.1 Hebephrenic schizophrenia
    - F20.2 Catatonic schizophrenia
    - F20.3 Undifferentiated schizophrenia
    - F20.4 Post-schizophrenic depression
    - F20.5 Residual schizophrenia
    - F20.6 Simple schizophrenia
    - F20.8 Other schizophrenia (Cenesthopathic schizophrenia)
    - F20.9 Schizophrenia, unspecified

2) Max ICD10 by type

```
sums = df.groupby(['icd10'])['cost'].sum()
sums.sort(inplace=True,ascending=False)
sums.head()
```

Code	Cost
I22	17380.7
Z51	13958.6
F20	11575.7

Table 3: Most expensive by Code sum

Table 3 show the most expensive codes by total. Since we already saw `I22,F20` I'll just put deases names for `Z51`

Z51 Other medical care

- Z51.0 Radiotherapy session
- Z51.1 Chemotherapy session for neoplasm
- Z51.2 Other chemotherapy

- Z51.3 Blood transfusion (without reported diagnosis)
- Z51.4 Preparatory care for subsequent treatment, not elsewhere classified
- Z51.5 Palliative care
- Z51.6 Desensitization to allergens
- Z51.8 Other specified medical care
- Z51.9 Medical care, unspecified