Supplementary

1. More details about the TF-IDF features for the three cases in experiments

Due to the extensive vocabulary across the entire dataset, the TI-IDF feature has 14,713 dimensions. For a case, it is a sparse vector with dimension 1x14713, where the value was not zero which means the word in this position appeared in the case.

To present more details, we provided the non-zero values and their positions in the supplement files for the cases in Section 4. To read the features, e.g. for (0, 574) 0.06732040735669434， the first number in bracket means the index of this case (is 0 due to the case itself), and the second one in bracket is the column index of non-zero features, the third number is the TF-IDF features.

Case 1:

```
 (0, 574)        0.06732040735669434
 (0, 5405)       0.10576706466561243
 (0, 9038)       0.08610749505245838
 (0, 5170)       0.09569839089545759
 (0, 5406)       0.09525188029959793
 (0, 5221)       0.07406662559788914
 (0, 10517)      0.1262840772126576
 (0, 10372)      0.15557784574621908
 (0, 9878)       0.1252340511819256
 (0, 11489)      0.06683309163298616
 (0, 49)         0.21178926125725903
 (0, 14039)      0.14768872334387856
 (0, 13743)      0.13156551001322533
 (0, 12606)      0.0454370354773355
 (0, 492)        0.09113856011428853
 (0, 7798)       0.11626688819552425
 (0, 14556)      0.12709819000901476
 (0, 11809)      0.19528131691147857
 (0, 9393)       0.08649399094433408
```

(0, 4049)	0.15071152246553407
(0, 4792)	0.14712133182592493
(0, 6474)	0.12161295836618408
(0, 45)	0.15071152246553407
(0, 8085)	0.12276887974152789
(0, 7626)	0.12541392316691072
(0, 2161)	0.14107994488249723
(0, 11293)	0.1563473438788523
(0, 13250)	0.23142146578985442
(0, 3078)	0.10169660346465156
(0, 3297)	0.18964549549816037
(0, 1019)	0.09446562576549968
(0, 4002)	0.04063913528001961
(0, 14271)	0.19858924462334054
(0, 6171)	0.1809348176268583
(0, 7635)	0.19889722006039262
(0, 5181)	0.058939368538538005
(0, 6455)	0.16658831834223808
(0, 10007)	0.09455190824475583
(0, 10515)	0.1380756409376103
(0, 4051)	0.1429529671273129
(0, 3086)	0.03076170415092282
(0, 13996)	0.09534061631309565
(0, 8855)	0.1282217368275654
(0, 5160)	0.22613576054187928
(0, 11570)	0.39056263382295714
(0, 7153)	0.18495379407479773

Case 2:

(0, 2766)	0.07296108351479014
(0, 2345)	0.07296108351479014
(0, 12923)	0.14592216702958027
(0, 9443)	0.14592216702958027
(0, 13536)	0.07296108351479014
(0, 9389)	0.07296108351479014

(0, 6344)	0.07296108351479014
(0, 9444)	0.6566497516331112
(0, 714)	0.06695128908798458
(0, 2533)	0.06206841891592062
(0, 5101)	0.14286538044460279
(0, 9340)	0.059216071132587554
(0, 6341)	0.22190090005958388
(0, 13133)	0.06323622954035579
(0, 3875)	0.06789628290625425
(0, 12092)	0.048462079152257705
(0, 10268)	0.19826573197106653
(0, 10334)	0.06323622954035579
(0, 1414)	0.05768767784009882
(0, 8522)	0.08118785609253465
(0, 692)	0.19368055209360033
(0, 10967)	0.10387763539769765
(0, 10842)	0.0490370187875673
(0, 1100)	0.059216071132587554
(0, 10638)	0.03986366455931742
(0, 9213)	0.05213912613984183
(0, 10773)	0.04918611829801376
(0, 8804)	0.056363723349254505
(0, 9373)	0.06387612449848601
(0, 5108)	0.04131414051092594
(0, 9255)	0.13058991886407836
(0, 10491)	0.0980740375751346
(0, 7865)	0.0855497096669221
(0, 3046)	0.17270529548483277
(0, 12832)	0.047403482085568695
(0, 13439)	0.056678509489695994
(0, 2841)	0.04644708524062478
(0, 5191)	0.0417302126326934
(0, 4571)	0.05576263443127024
(0, 11761)	0.162453811572155
(0, 1532)	0.20619978819934645
(0, 13529)	0.05343421618351606
(0, 11966)	0.037852717558189815
(0, 9396)	0.05081517164899752
(0, 13283)	0.03110068588214001
(0, 9214)	0.0498060032985947
(0, 7528)	0.040980272920052435
(0, 13132)	0.05881132388995011
(0, 9683)	0.03110068588214001
(0, 6612)	0.04254155158464653

(0, 2595)	0.05969945015017831
(0, 2344)	0.07804105636525001
(0, 11025)	0.05576263443127024
(0, 3887)	0.04667767541496611
(0, 8757)	0.02897099996044789
(0, 5068)	0.10271990328653975
(0, 11666)	0.030790446137162388
(0, 497)	0.031052320271109626
(0, 12797)	0.035000369774856765
(0, 914)	0.037710054224925155
(0, 10681)	0.03547873378108789
(0, 11108)	0.06695128908798458
(0, 3669)	0.044869969832400707
(0, 3056)	0.026739010319859046
(0, 9577)	0.08508310316929306
(0, 4752)	0.05013111211628343
(0, 13130)	0.11256892353753849
(0, 6921)	0.07523214627287415
(0, 2373)	0.03829345369914423
(0, 9066)	0.035746204775950634
(0, 4489)	0.05046724364839303
(0, 11597)	0.05234356494148627
(0, 2193)	0.0326169656354306
(0, 7270)	0.05713719944941978
(0, 4569)	0.03614127419390523
(0, 6154)	0.018985592670299152
(0, 14078)	0.041944984869979156
(0, 2701)	0.03804618575703475
(0, 12607)	0.03963172578179115
(0, 12041)	0.03180381092839494
(0, 4932)	0.054924253797222544
(0, 741)	0.03310919465037057
(0, 12428)	0.0317260463681824
(0, 327)	0.019379036099560938
(0, 2321)	0.06270424678764064
(0, 719)	0.02644605365707302
(0, 9145)	0.028613611587423895
(0, 5170)	0.06756955251973158
(0, 9393)	0.09160593311551349
(0, 13250)	0.040849811424632475
(0, 4002)	0.04304097738685335
(0, 7635)	0.035108730750710096
(0, 5181)	0.041615190398500125
(0, 3086)	0.05429962209325101

Case 3:

(0, 13874)	0.114674096946384
(0, 7076)	0.10788421073672567
(0, 6266)	0.04899704573249701
(0, 2353)	0.040238459122134314
(0, 11091)	0.05744111977179295
(0, 12954)	0.03552446530840264
(0, 190)	0.08466943645192082
(0, 12190)	0.09024824665796435
(0, 14544)	0.05157291487005549
(0, 3616)	0.08875873918467007
(0, 9145)	0.02416362081038293
(0, 8738)	0.16564311345795807
(0, 2383)	0.05332438718873705
(0, 10520)	0.11422978047313825
(0, 4604)	0.12582349542758114
(0, 14190)	0.039451399983267466
(0, 823)	0.05811328705582998
(0, 3408)	0.042666594638788045
(0, 722)	0.04391634221263599
(0, 14316)	0.12628148408878642
(0, 1282)	0.07048065437752996
(0, 5232)	0.25256296817757284
(0, 189)	0.155759100484525
(0, 11035)	0.09322368141403861
(0, 9640)	0.21835575283070716
(0, 13817)	0.23682168195151534
(0, 10322)	0.10680345383335524
(0, 5062)	0.09468076696071322
(0, 13953)	0.24129389395302672
(0, 13403)	0.07406854480574336
(0, 6479)	0.30860540816907056
(0, 3763)	0.08840618048129449
(0, 14656)	0.04616644396837938
(0, 13384)	0.08466943645192082
(0, 7870)	0.09145932266157913
(0, 8205)	0.09188329264531803
(0, 8981)	0.07825587888556668
(0, 9902)	0.20785132640312473
(0, 2258)	0.3221214168845915
(0, 13322)	0.053281503887791105
(0, 12592)	0.4411222639895151

(0, 7219)	0.2696714581018085
(0, 574)	0.04014047218446321
(0, 5405)	0.06306467955173715
(0, 12606)	0.054184581773511466
(0, 7626)	0.07477931718018915
(0, 3086)	0.01834197650163233

2. More details about the model computation

As shown in Fig.9, there are several 1D convolutional layers in the deep networks, in our work, we selected 3 layers. Each layer has 3 basic operations: 1D convolution operation, activation function, and maxpooling.

1) First step: it entered the convolutional operation, we represent it in general as follows:

$$\boldsymbol{y}^{\ l} = \sum_{k=1}^{N_k} \left[\boldsymbol{b}_k{}^{l} + conv1D(\boldsymbol{w}^{l}{}_{k}\ ,\boldsymbol{S}^{l-1}\ )\right]$$

The explained details are written in the paper.

2) Second step: The output of the extracting feature $\boldsymbol{y}$ will be as the input of the activation function. We used the ReLU as the activation function, and its format is as follows:

$$\boldsymbol{x}^{\ l} = f(\boldsymbol{y}^{l}\ ) \qquad \text{where } f(x) = max(0,x).$$

That means if it is positive, the function will output the input directly, otherwise, it will output zero.

3) Third Step: max-pooling is employed for the down-sampling.

$$\boldsymbol{S}^{l} = \boldsymbol{x}^{l} \downarrow \boldsymbol{s}$$

That means only keeping the maximum value in a rectangular window of size $\boldsymbol{s}$.

4) Finally, the output of the last layers, the output is flattened to the dense layer for the classification, represented as $\boldsymbol{z}$, and computed in the SoftMax function.

$$softmax(\ \boldsymbol{z}_i) = \frac{exp(\boldsymbol{z}_i)}{\sum_j \ exp(\boldsymbol{z}_j)}$$

The SoftMax function returns the probability of each class.