Project 4 - Example Main Script

Team 11

3/22/2017

Summary:

In this project, we implemented, evaluated and compared the algorithms in paper 1 : Information Processing and Management (Kang 2009), and paper 5: Author Disambiguation using Error-driven Machine Learning with a Ranking Loss Function(Culotta 2007) for Entity Resolution. We created an author disambiguation system that divides the same-name author occurrences in citation data into different clusters, each of which are expected to correspond to a real individual. Paper 1 suggested a single-link agglomerative clusering algorithm and each name occurrence is represented by a set of his/her coauthor names. Specifically, we set the number of overlapping coauthors to 1. We also used hierarchical clustering. In addition, we implemented Cluster Scoring Function, Error-driven Online Training, and Ranking MIRA. After comparing these two methods, ...

Step 0: Load the packages, specify directories

```
if (!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```
pacman::p_load(text2vec, plyr,qlcMatrix, kernlab, knitr)
setwd("~/Spr2017-proj4-team-11/doc")
```

Step 1: Load and process the data

For each record in the dataset, there are some information we want to extract and store them in a regular form: canonical author id, coauthors, paper title, publication venue title.

After generated a list of 14 elements using Professor Zheng's code, we reorganizes it into a list of 14 dataframes for easier access and processing.

```
source("../lib/dataclean.R")
```

Step 2: Feature design

Feature design

Paper 1 : Following the section 5.2, that each name occurrence is represented by a set of his/her coauthor names.

Paper 5: We want to use coauthors, paper titles and journey titles to design features for citations. We count the same-coauthor occurrences, and used bigram, trigram, TF-IDF, edit distance... to extract features from paper title and journey title.

```
# source("../lib/coauthormatrix.R")
load("../data/sim_matrix.RData")
```

Step 3: Clustering

We used a hierarchical clustering method for both paper 1 and paper 5. The algorithm also follows section 5.2 in paper 1.

Algorithm 2

Agglomerative Clustering for Same-name Author Occurrences

Input:	
	$a_1,\ldots,a_n$; same-name author occurrences
	$a_i = \{v_{i1},\ldots,v_{im}\}$; each name occurrence a_i has a set of m ($m \geqslant 0$) his/her coauthor names
	θ; a cluster-merging threshold
Initialize:	
	$c_i = \{a_i\}$; consider each name occurrence a_i as an element of cluster c_i
Loop:	
1	DO
2	For each cluster-pair (c_i, c_j), calculate $CSim(c_i,c_j)$
3	$CSim(c_i,c_j) = \max(ASim(a_x, a_y))$, $\forall a_x \in c_i$, $\forall a_y \in c_j$
4	$ASim(a_x,a_y) = \lvert a_x \cap a_y \rvert$
5	Find the most similar cluster-pair (c_u,c_v)
6	$(c_u,c_v) = \text{argmax}\ CSim(c_i, c_j)$
7	IF $CSim(c_u,c_v) \geqslant \theta$ THEN
8	$c_{u\_v} = c_u \cup c_v$; merge c_u and c_v into a new larger cluster c_{u_v}
9	ENDIF
10	WHILE ($CSim(c_u,c_v) \geqslant \theta$)
Output:	
	Clusters of author occurrences: $\{c_k\}$

For paper 1, we considered two scenarios : we combine all the single-element cluster; or we don't combine them.

```
source("../lib/singlelink.R")
start.time <- Sys.time()
# cluster_temp.list <- NULL
# cluster_temp.list <- llply(simmatrix.list,singlecluster,theta=1)
load("../doc/cluster_temp_1.RData")
cluster.combined <- NULL
cluster.combined <- llply(cluster_temp.list,combinecluster)
cluster.notcombined <- NULL
cluster.notcombined <- llply(cluster_temp.list,splitcluster)
end.time <- Sys.time()
time_scluter <- end.time - start.time
# combind cluster table for AGupta
table(cluster.combined[[1]])
```

```
##
##   0   1   2   3   4
## 569   2   2   2   2
```

```r
# do not combine single-element cluster
table(cluster.notcombined[[1]])
```

```
##
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   2   2   2   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450
```

```
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
```

Here, I only showed the cluster result of a subset of the data, "AGupta.txt" to further illustrate the difference of these two scenarios.

Evlatuion

Step 4: Evaluation

To evaluate the performance of the method, it is required to calculate the degree of agreement between a set of system-output partitions and a set of true partitions. In general, the agreement between two partitioins is measured for a pair of entities within partitions. The basic unit for which pair-wise agreement is assessed is a pair of entities (authors in our case) which belongs to one of the four cells in the following table (Kang et at.(2009)):

Matching matrix for the agreement between two sets of clusters

		Gold standard clusters (G)	
		Match	Mismatch
Machine-generated clusters (M)	Match	a	b
	Mismatch	c	d

Let M be the set of machine-generated clusters, and G the set of gold standard clusters. Then. in the table, for example, a is the number of pairs of entities that are assigned to the same cluster in each of M and G. Hence, a and d are interpreted as agreements, and b and c disagreements. When the table is considered as a confusion matrix for a two-class prediction problem, the standard "Precision", "Recall","F1", and "Accuracy" are defined as follows.

$$\text{Precision} = \frac{a}{a+b}$$
$$\text{Recall} = \frac{a}{a+c}$$
$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
$$\text{Accuracy} = \frac{a+d}{a+b+c+d}$$

```
source("../lib/evaluation_measures.R")

#### paper 1
matching_matrix_single <- NULL
matching_matrix_combined <- NULL
for (i in 1:14){
  matching_matrix_single[[i]] <- matching_matrix(data[[i]],cluster.notcombined[[i]])
  matching_matrix_combined[[i]] <- matching_matrix(data[[i]],cluster.combined[[i]])
}

f1.list.single <- NULL
accuracy.list.single <- NULL
f1.list.combined <- NULL
accuracy.list.combined <- NULL
clustering_errors_single <- NULL
clustering_errors_combined <- NULL
for (i in 1:14){
  f1.list.single[i] <- performance_statistics(matching_matrix_single[[i]])$f1
  f1.list.combined[i] <- performance_statistics(matching_matrix_combined[[i]])$f1
  accuracy.list.single[i] <- performance_statistics(matching_matrix_single[[i]])$accuracy
  accuracy.list.combined[i] <- performance_statistics(matching_matrix_combined[[i]])$accuracy
}
```

Step 3: Clustering

Following suggestion in the paper, we carry out spectral clustering on the Gram matrix of the citation vectors by using R function `specc()` in *kernlab*. The number of clusters is assumed known as stated in the paper.

Step 4: Evaluation

To evaluate the performance of the method, it is required to calculate the degree of agreement between a set of system-output partitions and a set of true partitions. In general, the agreement between two partitioins is measured for a pair of entities within partitions. The basic unit for which pair-wise agreement is assessed is a pair of entities (authors in our case) which belongs to one of the four cells in the following table (Kang et at.(2009)):

Matching matrix for the agreement between two sets of clusters

		Gold standard clusters (G)	
		Match	Mismatch
Machine-generated clusters (M)	Match	a	b
	Mismatch	c	d

Let M be the set of machine-generated clusters, and G the set of gold standard clusters. Then. in the table, for example, a is the number of pairs of entities that are assigned to the same cluster in each of M and G. Hence, a and d are interpreted as agreements, and b and c disagreements. When the table is considered as a confusion matrix for a two-class prediction problem, the standard "Precision", "Recall","F1", and "Accuracy" are defined as follows.

$$\text{Precision} = \frac{a}{a+b}$$
$$\text{Recall} = \frac{a}{a+c}$$
$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
$$\text{Accuracy} = \frac{a+d}{a+b+c+d}$$