IR Homework#3
Text/Spoken Document Using SVM

Reference:
-Libsvm- A Library for Support Vector Machines, Chih-Chung Chang and Chih-Jen Lin
http://www.csie.ntu.edu.tw/~cjlin/libsvm/
Features Used for Summarization

• **Acoustic Features**
  – *PITCH*: Min/max/mean/difference pitch values of a spoken sentence
  – *ENERGY*: Min/max/mean/difference value of energy features of a spoken sentence
  – *CONFIDENCE*: Posterior probabilities

• **Lexical Features**
  – *BIGRAM_SCORE*: Normalized bigram language model scores
  – *SIMILARITY*: Similarity scores between a sentence and its preceding/following neighbor sentence
  – *NUM_NAME_ENTITIES*: Number of named entities (NEs) in a sentence

• **Other Features**
  – *POSITION*: Sentence position
  – *DURATION*: Duration of the preceding/current/following sentence
  – *R-VSM*: Relevance score obtained by using the VSM summarizer
  – *R-LSA*: Relevance score obtained by using the LSA summarizer
An Example for the Feature Sets of Sentences

- **PSTN_Acoustic_Fea_Text (PTSND20011107_1.txt)** 10 dim
  
  ```
  1 10 1 1.000000 2 0.611535 3 16.384585 4 0.827808 5 0.001128 6 0.655035 7 0.590136 8 1.834960 9 0.002641 10 8.000000
  2 10 1 0.500000 2 0.472137 3 16.466834 4 0.761306 5 0.003814 6 0.675514 7 0.594440 8 12.355190 9 0.003260 10 13.000000
  3 10 1 0.383333 2 0.615447 3 16.527532 4 0.762232 5 0.026254 6 0.736652 7 0.447178 8 11.578128 9 0.005480 10 10.000000
  4 10 1 0.280000 2 0.690202 3 16.688328 4 0.778751 5 0.017785 6 0.642387 7 0.583867 8 12.175820 9 0.002438 10 11.000000
  5 10 1 0.200000 2 0.676445 3 16.215265 4 0.754912 5 0.009893 6 0.833833 7 0.603759 8 17.752199 9 0.009719 10 8.000000
  ```

- **PSTN_Lexical_Fea_Text (PTSND20011107_1.txt)** 9 dim
  
  ```
  1 24
  2 7 5 1 0 0 0.000000 0.000000 1.000000
  3 5 7 19 3 1 1 0.004556 0.000000 1.000000
  4 19 5 19 3 2 2 0.008772 0.641470 0.000000
  5 19 19 10 3 4 2 0.008772 0.784019 0.641470
  ```

- **PSTN_Other_Fea_Spoken(PTSND20011107_1.txt)** 4 dim
  
  ```
  1 -3363.491699 -3404.175781 0.000001 0.380121
  2 -3346.184114 -3399.680522 0.010119 0.739739
  3 -3294.791260 -3248.675840 5.051671 1.917458
  4 -3285.574707 -3344.844238 4.304012 1.573014
  5 -3291.159912 -3345.663086 3.953468 1.736150
  ```
Feature collection by Acoustic, Lexical, or others from source files (.wav)

Feature Extraction for SVM Model Training and Testing

SVM Model Training and Classification (Using svmtrain and svmpredict)

Using ROUGE to measure it

Rank and Change the transcripts of sentences to word IDs for ROUGE

Extract SVM score we need
Step 1: Feature Extraction for SVM Model Training

• Accumulate the statistics from feature sets and convert them into the input format of the “svmtrain” and “svmpredict” tool.
Step 1: Feature Extraction for SVM Model Training (Example: Path Setup)

#define TRAINING_MODE 1 //是否有使用人工摘要
const char *Acoustic_Path = ".../PSTN_Acoustic_Fea_Text/";
   //Acoustic feature path
const char *Lexical_Path = ".../PSTN_Lexical_Fea_Text/";
   //Lexical feature path
const char *Unsupervised_Path = ".../PSTN_Unsupervised_Fea_Text/";
   //Unsupervised feature path
const char *List_Path = "./train_ds2.txt";
   //training data list path
const char *SUMMARYResult = "../Label/human_train_ds2";
   //參考的label檔案
const char *feaOutput = "../SVM_Data/train_0.2_text_Human.data";
   //產生出SVM output的路徑
#define LABEL_RATIO 0.2 //人工摘要比率
Step 2: SVM Model Training and Classification

- Using Libsvm to model training and classification

\[
\text{svmtrain} \ -b \ 1 \ \ldots/\text{Step1}\ _\text{Fe...}/\text{SVM...}/\text{step1\_generated.train.data}
\]

\[
\text{probability\_estimates} \quad \text{input train data generated by step 1}
\]

\[
\text{svmpredict} \ -b \ 1 \ \ldots/\text{Step1}\ _\text{Fe...}/\text{SVM...}/\text{step1.test.data} \quad \text{svmtrain\_generated.model}
\]

\[
\text{result1.txt} \quad \text{output result}
\]

\[
\text{probability\_estimates} \quad \text{svmtrain generated model}
\]

\[
\text{input test data generated by step 1}
\]
Step 2: How SVM Model Classify

- SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space.
- Default kernel function:

\[ K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \]
Step 3: Preparation for ROUGE Evaluation

- Extract the sentence scores computed by “svmpredict” and rank the sentences accordingly by these scores.

- Change the transcripts of sentences to word IDs (by looking up to a lexicon).
Step 4: ROUGE

- perform the ROUGE-2 evaluation
Introduction to Rouge

• ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a package for automatic evaluation of summaries.

• There are four different ROUGE measures:
  – ROUGE-N (N-gram Co-Occurrence Statistics)
  – ROUGE-L (Longest Common Subsequence)
  – ROUGE-W (Weighted Longest Common Subsequence)
  – ROUGE-S (Skip-Bigram Co-Occurrence Statistics)
Install perl and package

• Install perl
  – 安裝檔案為: ActivePerl-5.8.8.817-MSWin32-x86-257965.msi
    http://www.activestate.com/Products/activeperl/
  – 將ActivePerl-5.8.8.817-MSWin32-x86-257965.msi 打開安裝完成，如果沒有更改路徑，就會在C槽底下出現一個Perl的資料夾，Perl就已經安裝完成。
Install package

- 現在要安裝兩個package，這是為了之後執行ROUGE時，可以讀入xml檔以及讀入.db的資料庫檔。
  - 在開始功能表的程式集中找到ActivePerl 5.8.8 Build 817 裡面有Perl Package Manager 將此檔打開。
  - 接著會出現視窗畫面為
    ppm>
  - 安裝第一個package。回到ppm> 在>後面打入
    install XML-DOM
    它就會自行完成安裝。
  - 安裝第二個package。回到ppm> 在>後面打入
    install DB_file
  - 都安裝完成之後，就可以開始使用ROUGE了。
ROUGE-N

• ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries.

\[
\text{ROUGE} - N = \frac{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}} (\text{gram}_n)}{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{\text{gram}_n \in S} \text{Count} (\text{gram}_n)}
\]

\( S : \text{sentence} \)
\( n : \text{length of the n-gram} \)
Usage of ROUGE

• Before evaluation, prepare ...
  – Automatic generated summaries
  – Reference summaries by hand

• Command line argument
  – (-e): Directory which contains a database (.db) file
  – (-n): Length of the n-gram
  – (-a): A description file in XML format about the paths of the corresponding result files.

  – An example:
  perl ROUGE-1.5.5.pl -e dict_data -n 2 -a description file (XML)
An Example of the Description File

```xml
<ROUGE-EVAL version="1.0">
  <EVAL ID="1">
    <PEER-ROOT>
      summaryRatioFile/SUM/01
    </PEER-ROOT>
    <MODEL-ROOT>
      summaryRatioFile
    </MODEL-ROOT>
    <INPUT-FORMAT TYPE="SPL">
      Directory which contains automatic summaries
    </INPUT-FORMAT>
    <PEERS>
      <P ID="1">N200108011200-01.txt</P>
    </PEERS>
    <MODELS>
      <M ID="1">Human1/01/N200108011200-01.txt</M>
      <M ID="2">Human2/01/N200108011200-01.txt</M>
      <M ID="3">Human3/01/N200108011200-01.txt</M>
    </MODELS>
  </EVAL>
</ROUGE-EVAL>
```

Directory which contains reference summaries

File of automatic summary

1st reference summary
2nd reference summary
3rd reference summary