**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**TRẦN TẤN BẢO – 17C 11002**

**TRẦN THÚY HIỀN – 17C 11026**

**NGUYỄN HÀ DUY PHƯƠNG – 17C 11 032**

ĐỒ ÁN MÔN HỌC

XỬ LÝ NGÔN NGỮ NÓI

**BÀI TẬP 2: NHẬN DẠNG RỜI RẠC**

**GIÁO VIÊN**

**PGS. TS. Vũ Hải Quân**

**TP.HCM - 6/2018**

MỤC LỤC

1 Mô tả bài toán 2

2 Phương pháp làm 2

2.1 Các bước huấn luyện (mở cmd trỏ đến thư mục htk, lưu ý các file trong project này đều phải lưu dưới dạng ANSI): 2

2.2 Các bước test: 9

3 Kịch bản thử nghiệm 10

4 Kết quả, vẽ biểu đồ 10

4.1 Kết qủa thực nghiệm 10

4.2 Biểu đồ 11

5 Đánh giá cá nhân 11

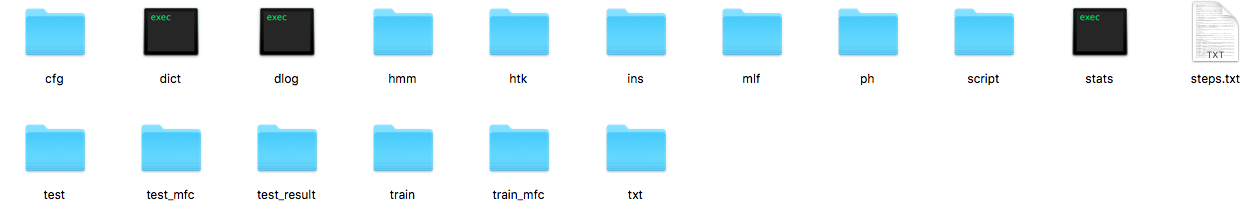
6 Tài liệu tham khảo 12

# Mô tả bài toán

* Mục tiêu của bài tập là nhận dạng các từ rời rạc, chẳng hạn nhận dạng chữ số từ không đến chín.
* Kiểm tra sự thay đổi chất lượng nhận dạng thông qua thay đổi kịch bản thực nghiệm.

# Phương pháp làm

* Sử dụng HTK Toolkit của Đại học Cambridge và mô hình HMM.
* Xây dựng n mô hình tương ứng với n âm vị.
* Tạo thư mục bài tập **htk** (Báo cáo giả định là thư mục chạy ngay trên ổ đĩa C)
* Trong thư mục **htk** có các mục sau: **bin32** chứa các chương trình con của HTK, **cfg** chứa các file config, **scripts** chứa các file Perl (cần cài đặt môi trường cho Perl), **wav** chứa hai thư mục con **train** và **test** để chứa các file \*.wav dùng để train và test tương ứng:
* Gồm có train: 4 file âm thanh đọc liên tục từ 0 đến 9 và test: 60 file âm thanh test rời rạc các chữ số.



## Các bước huấn luyện (mở cmd trỏ đến thư mục htk, lưu ý các file trong project này đều phải lưu dưới dạng ANSI):

|  |  |
| --- | --- |
| **Bước 1** | Tạo file **gram.txt** chứa các từ cần nhận dạng (ở bài tập này là từ 0 đến 9)    (Lưu ý có dấu xuống dòng ở cuối file, nếu không sẽ bị lỗi) |
| **Bước 2** | Tạo file **wdnet** bằng lệnh:  Htk/HParse.exe txt/gram.txt txt/wdnets |
| **Bước 3** | Tạo file dict và sắp xếp theo thứ tự alphabet:    (Lưu ý có dấu xuống dòng ở cuối file, nếu không sẽ bị lỗi) |
| **Bước 4** | Tạo file thu âm với số lượng train là 4 file, test là 60 file.  Ta đã tạo được file **train** chứa tên và nội dung của các file cần được ghi âm và đưa vào thư mục **train**.  Làm tương tự cho **test**.s |
| **Bước 5** | Chạy lệnh sau (lưu ý khi chạy các file Python cần có từ “Python” đứng đầu câu lệnh trong cmd):  Python script/listwavmfc.py train/ train\_mfc/ train.scp train\_mfc.scp  Lệnh này nhằm tạo ra file **script/listwavmfc.py**, file này nhằm tổng hợp các file cần rút trích đặc trưng mfcc trong **train** và tạo danh sách các file mfcc xuất ra tương ứng trong thư mục **train\_mfc**. |
| **Bước 6** | Rút trích đặc trưng mfcc:  htk/HCopy.exe -T 1 -C cfg\HCopy.cfg -S train\_mfc.scp  Ta sẽ có các file mfcc tương ứng với từng file train trong **train**.  Tạo file **train\_mfc.scp**, lưu danh sách các file mfcc trong **train** vào thư mục **train\_mfc**. Kết quả: |
| **Bước 7** | Tạo các monophone  Htk/HDMan.exe -m -w txt/wlist.txt -n ph/monophones.txt -l dlog txt/dict.txt txt/srcDict.txt  Perl script/mkMonophones.pl ph/monophones.txt ph/monophones0.txt ph/monophones1.txt  Sử dụng *monophones.txt* đã tạo ra để tạo thêm các *monophones0.txt* và *monophones1.txt*  Lần lượt là các monophones: *monophones.txt*, *monophones0.txt*, *monophones1.txt* khác nhau ở các **sil** và **sp**. |
| **Bước 8** | Tạo file **wlist** (word list):  Scripts/prompts2wlist.pl train txt/wlist |
| **Bước 9** | Tạo MLF *words*  perl script/prompts2mlf.pl mlf/words.mlf txt/prompts.txt |
| **Bước 10** | Tạo file **mkphone0.led**:    (Lưu ý có dấu xuống dòng ở cuối file) |
| **Bước 11** | Tạo các MLF *phones* từ MLF *words*.  htk/HLEd.exe -l "\*" -d txt/srcDict.txt -i mlf/phones0.mlf ins/mkphones0.led mlf/words.mlf  htk/HLEd.exe -l "\*" -d txt/srcDict.txt -i mlf/phones1.mlf ins/mkphones1.led mlf/words.mlf |
| **Bước 12** | Tạo thư mục **hmm/hmm0**, **hmm/hmm1**, …, **hmm/hmm22**  Tạo file proto mẫu theo htkbook, copy vào **hmm/hmm0**, hoặc tạo file proto với nội dụng sau và copy vào **hmm/hmm0:** |
| **Bước 13** | Khởi tạo giá trị train:  htk/HCompV.exe -C htk/cfg/HCompV.cfg -f 0.01 -m -S txt/train.scp -M hmm/hmm0 hmm/hmm0/proto |
| **Bước 14** | Tạo file **hmmdefs**:  Perl script/mkHmmdefsFile.pl hmm/hmm0/proto ph/monophones0.txt hmm/hmm0/hmmdefs |
| **Bước 15** | Tạo file **macros**:  perl script/mkMacrosFile.pl hmm/hmm0/vFloors hmm/hmm0/macros |
| **Bước 16** | Xây dựng mô hình **hmm1** và **hmm2**  htk/HERest.exe -C cfg/HERest.cfg -I mlf/phones0.mlf -t 250.0 150.0 1000.0 -S txt/train\_mfc.scp -H hmm/hmm0/macros -H hmm/hmm0/hmmdefs -M hmm/hmm1 ph/monophones0.txt  htk/HERest.exe -C cfg/HERest.cfg -I mlf/phones0.mlf -t 250.0 150.0 1000.0 -S txt/train\_mfc.scp -H hmm/hmm1/macros -H hmm/hmm1/hmmdefs -M hmm/hmm2 ph/monophones0.txt |
| **Bước 17** | Thực hiện tương tự cho huấn luyện ***hmm4***, ***hmm5***. |
| **Bước 18** | Thêm mô hình **sp** vào **hmmdefs4**  perl script/makesp.pl hmm/hmm3/hmmdefs hmm/hmm4/hmmdefs hmm/hmm3/macros hmm/hmm4/macros |
| **Bước 19** | Sử dụng **HHed** để trói buộc mô hình **sil** và **sp**, huấn luyện **hmm5**  htk/HHEd.exe -H hmm/hmm4/macros -H hmm/hmm4/hmmdefs -M hmm/hmm5 ins/sil.led ph/monophones1.txt  File sil.led có cấu trúc như sau:    Huấn luyện tiếp **hmm6**, **hmm7** tương tự bước 14, tuy nhiên ở đây lúc này sử dụng *monophones1.txt* chứ không dùng *monophones0.txt* như ở trên. |
| **Bước 20** | Tạo ra file **aligned.mlf** từ **phones.mlf**  htk/HVite.exe -l \* -o SWT -b silence -a -H hmm/hmm7/macros -H hmm/hmm7/hmmdefs -i mlf/aligned.mlf -m -t 250.0 -y lab -I mlf/words.mlf -S txt/train\_mfc.scp txt/srcDict.txt ph/monophones1.txt |
| **Bước 21** | Huấn luyện tiếp tục **hmm8**, **hmm9** từ **aligned.mlf** có được bằng lệnh  htk/HERest.exe -B -C cfg/HERest.cfg -I mlf/aligned.mlf -t 250.0 150.0 1000.0 -s stats -S txt/train\_mfc.scp -H hmm/hmm7/macros -H hmm/hmm7/hmmdefs -M hmm/hmm8 ph/monophones1.txt  htk/HERest.exe -B -C cfg/HERest.cfg -I mlf/aligned.mlf -t 250.0 150.0 1000.0 -s stats -S txt/train\_mfc.scp -H hmm/hmm8/macros -H hmm/hmm8/hmmdefs -M hmm/hmm9 ph/monophones1.txt |
| **Bước 22** | Tạo **triphones** từ monophones bằng lệnh  htk/HLEd.exe -n ph/triphones1 -l \* -i mlf/wintri.mlf ins/mktri.led mlf/aligned.mlf |
| **Bước 23** | Tạo **mktri.hed** để huấn luyện **hmm10** là nhái monophones từ **hmm9**  htk/HHEd.exe -B -H hmm/hmm9/macros -H hmm/hmm9/hmmdefs -M hmm/hmm10 ins/mktri.hed ph/monophones1.txt  **Hmm11**: để ước lượng triphones.  htk/HERest.exe -B -C cfg/HERest.cfg -I mlf/wintri.mlf -t 250.0 150.0 1000.0 -s stats -S txt/train\_mfc.scp -H hmm/hmm10/macros -H hmm/hmm10/hmmdefs -M hmm/hmm11 ph/triphones1  Tiếp tục huấn luyện **hmm12**.  htk/HERest.exe -B -C cfg/HERest.cfg -I mlf/wintri.mlf -t 250.0 150.0 1000.0 -s stats -S txt/train\_mfc.scp -H hmm/hmm11/macros -H hmm/hmm11/hmmdefs -M hmm/hmm12 ph/triphones1 |
| **Bước 24** | Tạo **fulllist** từ monophones0  perl script/mkFullList.pl ph/monophones0.txt ph/fulllist |
| **Bước 25** | Tạo **tree.hed**  perl script/mkTree.pl 40 ph/monophones0.txt ins/tree.hed  Vào **tree.hed**, sửa lại đường dẫn muốn lưu *fullist*, *tiedlist* và *tree* |
| **Bước 26** | Tạo trạng thái ràng buộc cho **triphones**. Đây là bước quan trọng và tốn thời gian nếu tập dữ liệu quá lớn.  htk/HHEd.exe -B -H hmm/hmm12/macros -H hmm/hmm12/hmmdefs -M hmm/hmm13 ins/tree.hed ph/triphones1 > txt/log.txt  Sau bước này, **tiedlist** và **tree** sẽ được tạo ra theo đường dẫn lưu đã chọn ở |
| **Bước 27** | Cuối cùng, huấn luyện **hmm14**, **hmm15** với **tiedlist** đã được tạo ra ở bước trên.  htk/HERest.exe -B -C cfg/HERest.cfg -I mlf/wintri.mlf -t 250.0 150.0 1000.0 -s stats -S txt/train\_mfc.scp -H hmm/hmm13/macros -H hmm/hmm13/hmmdefs -M hmm/hmm14 ph/tiedlist  htk/HERest.exe -B -C cfg/HERest.cfg -I mlf/wintri.mlf -t 250.0 150.0 1000.0 -s stats -S txt/train\_mfc.scp -H hmm/hmm14/macros -H hmm/hmm14/hmmdefs -M hmm/hmm15 ph/tiedlist  **=> Đến đây là xong các bước để train dữ liệu.** |

## Các bước test:

|  |  |
| --- | --- |
| **Bước 1** | * Tạo thư mục **test\_result** để lưu kết quả test |
| **Bước 2** | * Thu âm các file test và bỏ vào thư mục **test**, và tạo tập tin **prompts.txt** tương ứng các từ trong file âm thanh.      * Tạo file **test.scp** và **test\_mfc.scp:**   python script/listwavmfc.py test/ test\_mfc/ txt/test.scp txt/test\_mfc.scp   * Rút trích đặc trưng mfcc của các file test:   htk/HCopy.exe -T 1 -C cfg/HCopy.cfg -S txt/test.scp |
| **Bước 3** | * Tạo file **words.mlf** lưu trong thư mục **test\_result**:   perl script/prompts2mlf.pl test\_result/words.mlf test\_result/prompts.txt |
| **Bước 4** | Thực hiện test dữ liệu:  htk/HVite.exe -C cfg/Hvite.cfg -H hmm/hmm15/macros -H hmm/hmm15/hmmdefs -S txt/test\_mfc.scp -i test\_result/recout.mlf -w txt/wdnet.txt txt/srcDict.txt ph/tiedlist |
| **Bước 5** | Thống kê kết quả, xuất kết quả ra tập tin txt.  htk/HResults.exe -f -t -I test\_result/words.mlf ph/tiedlist test\_result/recout.mlf > test\_result/result.txt  Đây là kết quả của lần test dùng hmm15.      Thực hiện kiểm thử trên hmm14, hmm13 để có các kết quả khác. |

# Kịch bản thử nghiệm

* Thử nghiệm với đặc trưng MFCC\_0\_D\_A

Trong đó: MFCC là đặc trưng bố cục khoang miệng khi phát âm, \_0 là độ lớn của vector, \_D là đạo hàm bậc 1 (đặc trưng tương quan theo thứ tự), \_A là đạo hàm bậc 2 (đặc trưng tương quan theo thời gian). \_X nghĩa là kết hợp MFCC với đặc trưng X (X là 0, D, A).

* Tiến hành kiểm thử qua 3 hmm: **hmm13**, **hmm14**, **hmm15**.

# Kết quả, vẽ biểu đồ

## Kết qủa thực nghiệm

Tiến hành kiểm thử qua 3 hmm: **hmm13**, **hmm14**, **hmm15**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Đặc trưng**  **MFCC\_0\_D\_A**  **Với các hmm** | **Độ chính xác** | **Tổng số từ** | **Số từ đúng** | **Lỗi chèn** |
| Hmm13 | 25% | 60 | 15 | 40 |
| Hmm14 | 25% | 60 | 15 | 40 |
| Hmm15 | 23.33% | 60 | 14 | 42 |

## Biểu đồ

Biểu đồ đánh giá thay đổi độ chính xác qua các hmm

# Đánh giá cá nhân

* Việc kết hợp MFCC và các đặc trưng khác như đạo hàm bậc 1, đạo hàm bậc 2 làm tăng khả năng nhận dạng của mô hình.
* Chưa thực hiện kiểm thử được cho các kịch bản MFCC\_0\_D, MFCC\_0\_A, …; thay đổi số vòng lặp huấn luyện, …
* Việc sử dụng HTK đòi hỏi nhiều thời gian nghiên cứu nên các thông số cũng như các hàm trong các bước trên chưa được mô tả rõ ràng và có thể phát sinh lỗi.
* **Vấn đề:** Chưa thực hiện chạy được toàn bộ tập dữ liệu vivos. Vấn đề xảy ra vì tập dữ liệu quá lớn, khi thực hiện để bước huấn luyện hmm13 thì quá mất thời gian. Nên nhóm dùng 4 tập tin train và 60 tập tin test với mục đích thể hiện hiểu và thực hiện được các bước của quá trình train.

# Tài liệu tham khảo

* Các bước thực hiện huấn luyện và kiểm thử phía trên được tham khảo chủ yếu (có chỉnh sửa, thêm hình ảnh và giải thích) từ file “**DauTai\_HuongDanHTK\_2014\_HCMUS.docx**” có trong thư mục **ref**
* Ngoài ra còn tham khảo ở htk\_training\_vn.pdf và htkbook.pdf