

**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 TIẾNG NÓI LIÊN TỤC**

**GIÁO VIÊN**

**PGS. TS. Vũ Hải Quân**

**TP.HCM - 6/2018**

# MỤC LỤC

<b>1</b>	<b>Mô tả bài toán .....</b>	<b>2</b>
<b>2</b>	<b>Phương pháp làm.....</b>	<b>2</b>
2.1	Tổng quát cách làm.....	2
2.2	Các bước huấn luyện (mở cmd trở đến thư mục htk, lưu ý các file trong project này đều phải lưu dưới dạng ANSI):.....	4
2.3	Các bước test: .....	5
<b>3</b>	<b>Kịch bản thử nghiệm .....</b>	<b>8</b>
<b>4</b>	<b>Kết quả, vẽ biểu đồ .....</b>	<b>9</b>
4.1	Kết quả thực nghiệm.....	9
4.2	Đánh giá kết quả .....	9
<b>5</b>	<b>Đánh giá cá nhân .....</b>	<b>10</b>
<b>6</b>	<b>Tài liệu tham khảo .....</b>	<b>10</b>

## 1 Mô tả bài toán

- Mục tiêu của bài tập là nhận dạng các câu nói liên tục, có thể là chữ số hay nội dung bất kì.
- Kiểm tra sự thay đổi chất lượng nhận dạng thông qua xây dựng Language Model bigram và trigram.

## 2 Phương pháp làm

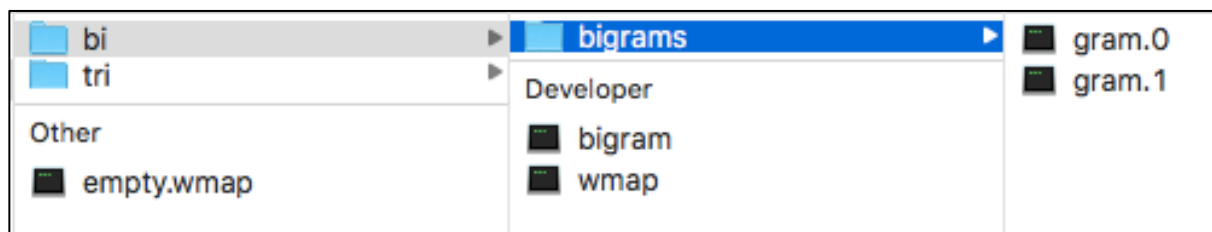
### 2.1 Tổng quát cách làm

Sử dụng tiếp tục kết quả bài 2 (nhận dạng rời rạc)s

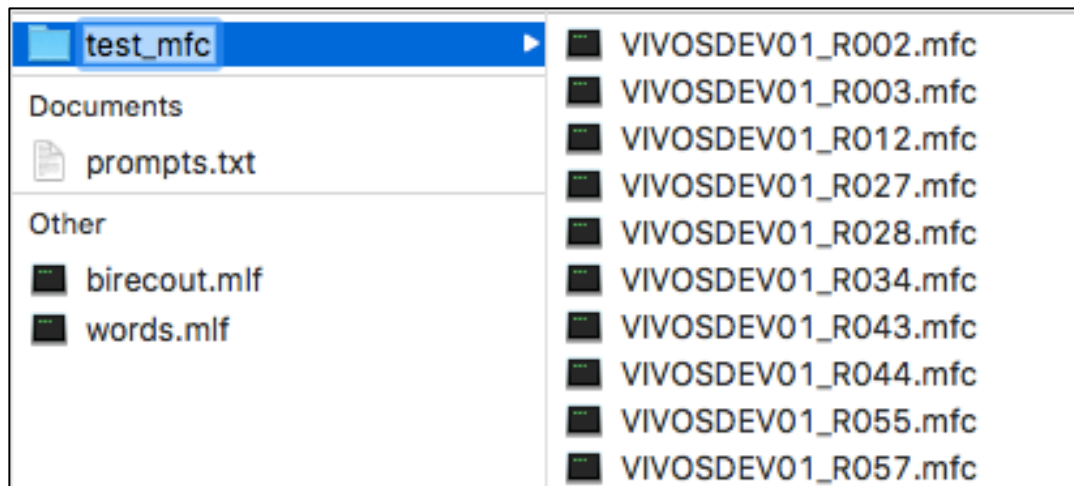
- Sử dụng HTK Toolkit của Đại học Cambridge và mô hình HMM.
- Tạo thư mục bài tập **htk** (Báo cáo giả định là thư mục chạy ngay trên ổ đĩa C)
- Cài đặt thêm môi trường **perl** và **python** cho một số bước phát sinh tập tên file...

Tổ chức dữ liệu:

- Tập dữ liệu huấn luyện gồm:
  - o *Train*: gồm tất cả file âm thanh **.wav** có trong thư mục *vivos/train*.
  - o *Test*: gồm tất cả file âm thanh **.wav** trong thư mục *vivos/test*.
- Thêm thư mục:
  - o *gram*: gồm 2 thư mục con **bi** và **tri** chứa *bigram* và *trigram* tương ứng. Trong mỗi thư mục có thư mục *bigrams* chứa các file *gram.\**. Cấu trúc file như hình:

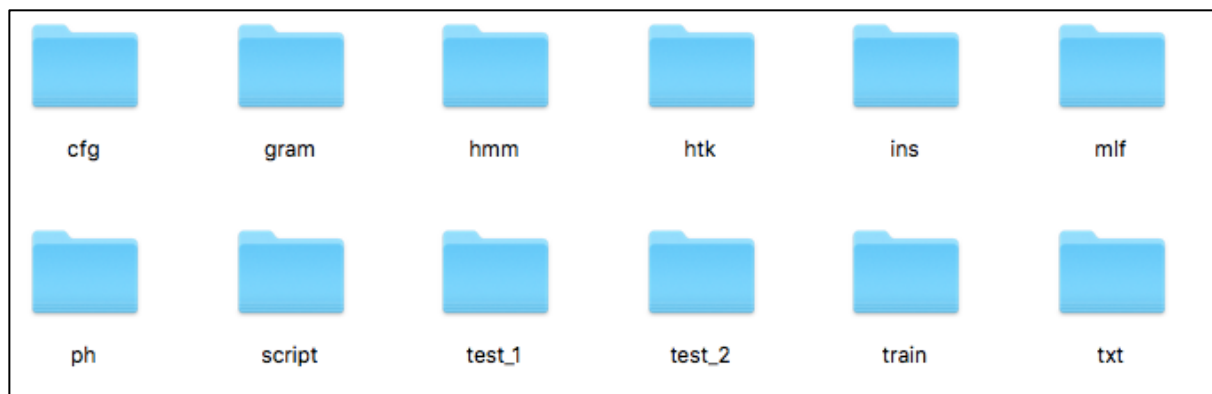


- o *test*: chứa các tập tin *prompts.txt* – dữ liệu chuỗi chữ thường, *words.mlf*, tập tin kết quả *birecout.mlf*, thư mục *test\_mfc* - danh sách file mfc phát sinh.



Cấu trúc thư mục **htk** có các mục sau:

- **cfg**: chứa các file config.
- **gram**: chứa language model bigram, trigram.
- **hmm**: chứa các thư mục hmm.
- **htk**: chứa các chương trình con của HTK.
- **ins**: chứa các file *.hed*, *.led*, *fulllist*, *tiedlist*.
- **mlf**: chứa các file *.mlf*.
- **ph**: chứa các file *phones*.
- **scripts**: chứa các file Perl (cần cài đặt môi trường cho Perl).
- **test\_1**: chứa tập tin kết quả *recout.mlf* và *result.txt* của bigram trên toàn bộ dữ liệu *vivos/test*, chứa thư mục *test\_mfc* – gồm tất cả các tập tin test *.mfc*.
- **test\_2**: chứa tập tin kết quả *recout.mlf* và *result.txt* của **bigram** và **trigram** trên 556 file dữ liệu *vivos/test*.
- **train**: tất cả tập tin train *.mfc*.
- **txt**: tập tin từ điển, và một số tập tin cấu hình khác.

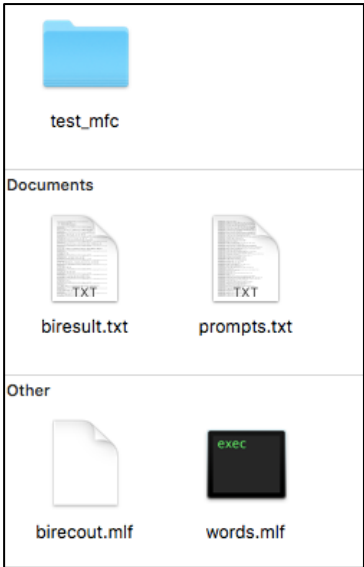
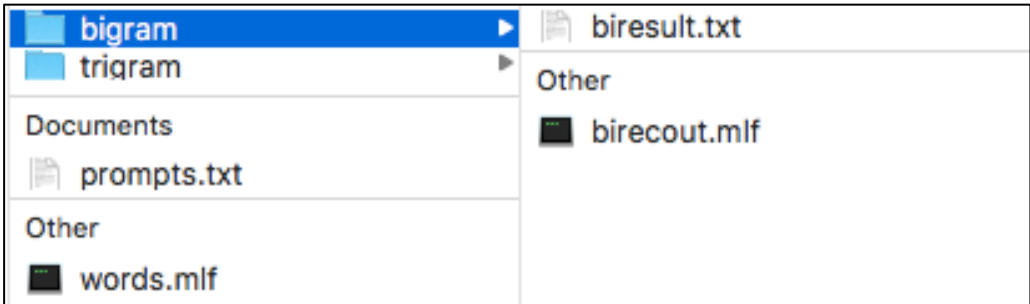


## 2.2 Các bước huấn luyện (mở cmd trở đến thư mục htk, lưu ý các file trong project này đều phải lưu dưới dạng ANSI):

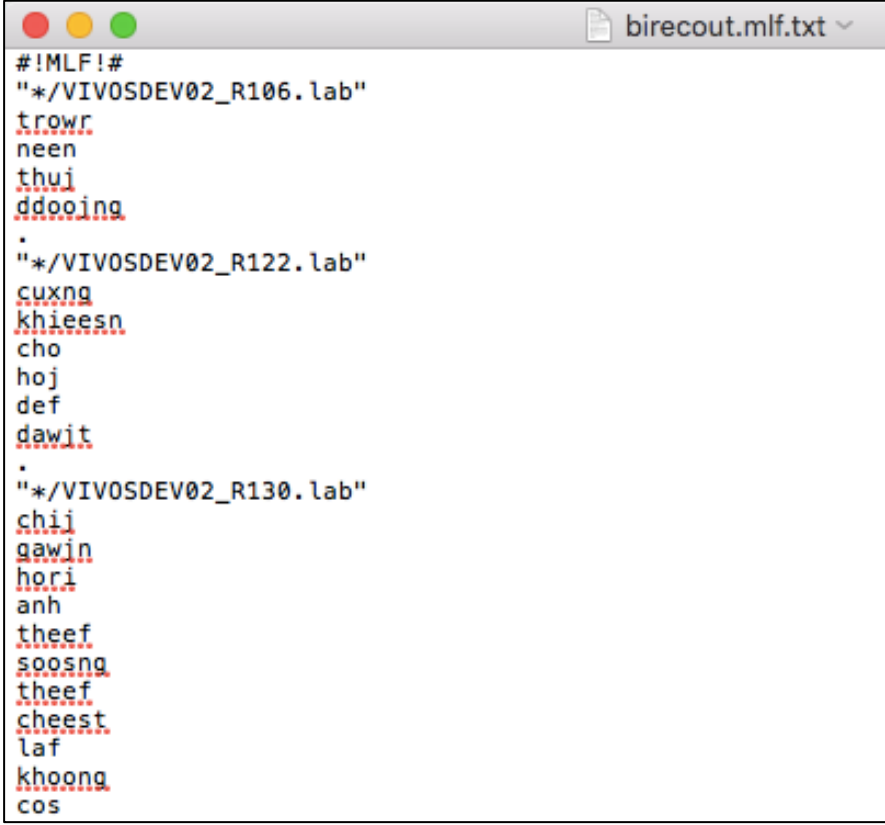
<b>Bước</b>  <b>1</b>	<p>Xây dựng hệ thống LVCSR – huấn luyện mô hình cross-word triphones.</p> <ul style="list-style-type: none"> <li>- Copy đề các file cấu hình trong thư mục đề bài tập cung cấp: <i>Hdecode.conf</i>, <i>fullist</i>, <i>mktri.hed</i>, <i>mktri.led</i>, <i>tree.hed</i> vào trong thư mục bài làm tương ứng.</li> <li>- Thực hiện chạy lại từ bước tạo triphones ở bài tập 2 với thông số thay đổi để tạo lại <i>tiedlist</i> và huấn luyện đến <i>hmm15</i>.</li> </ul>
<b>Bước</b>  <b>2</b>	<p>Xây dựng Language model</p> <ul style="list-style-type: none"> <li>- Tiền xử lý dữ liệu: Tạo tập tin <i>train.txt</i> gồm: dữ liệu prompts của tập tin âm thanh huấn luyện và browse bài báo web, lấy nội dung bài báo (5Mb)</li> <li>- Cấu trúc file theo dạng <code>&lt;s&gt; ... &lt;/s&gt;</code>, được kết quả:</li> </ul> <pre> &lt;s&gt; khasch sajn &lt;/s&gt; &lt;s&gt; chir bawfng casch luoon noox luwjc thif cuoosi cufng bajn mowsi dduwowjc ddeefn ddaasp &lt;/s&gt; &lt;s&gt; trong soos casc quoosc gia coong nghieejp phast trieern &lt;/s&gt; &lt;s&gt; anh ddax nhifn thaasy trong nhuwxng lari nhari doong dafi cura nhu &lt;/s&gt; &lt;s&gt; khurng hoarng mooi truwowfng caafn dduwowjc ngawn chawjn &lt;/s&gt; &lt;s&gt; troong thaasy tafu cas ddawfng xa anh ddax aasn cofi carnh baso &lt;/s&gt; &lt;s&gt; maasy safo ruoojng cura gia ddifnh oong ddafnh bor hoang cho cor mojc &lt;/s&gt; &lt;s&gt; cos ddeem gawjp phari heo nojc nos ruwowjt &lt;/s&gt; &lt;s&gt; casc bafn hojc nhor hawfng ngayf nay trowr thafnh bafn thowf cura casc chasu &lt;/s&gt; </pre>
<b>Bước</b>  <b>3</b>	<p>Chạy các lệnh sau để tạo <i>bigram</i> và <i>trigram</i>:</p> <ul style="list-style-type: none"> <li>- Tạo bảng map  <code>htk/LNewMap.exe -f WFC LMName gram/empty.wmap</code></li> <li>- Tạo wmap và gram.0  <code>htk/LGPrep.exe -T 1 -a 100000 -b 200000 -n 3 gram/empty.wmap txt/train.txt</code></li> <li>- Tạo bigram, trigram  <code>htk/LBuild.exe -T 1 -c 2 0 -c 3 0 -n 3 gram/tri/wmap trigram gram.0</code></li> <li>- Lượng giá huấn luyện. Sử dụng 1 file dữ liệu nhỏ để đánh giá bigram và trigram  <code>htk/LPlex.exe -n 3 -t gram/tri/trigram gram/test.txt</code></li> <li>- Tương tự, để thực hiện tạo <i>bigram</i> thì đổi <code>-n 3</code> thành <code>-n 2</code></li> </ul> <p>Kết quả lượng giá của <i>trigram</i></p>

	<pre> C:\Users\TuyenTran&gt;D:/Thacsi/NNN/htk/htk/LPlex.exe -n 3 -t D:/Thacsi/NNN/htk/gram/tri/trigram D:/Thacsi/NNN/htk/gram/ m/test_gram.txt LPlex test #0: 3-gram perplexity 1.#QNB, var -1.#IND, utterances 1, words predicted 0 num tokens 4, OOV 2, OOV rate 66.67% (excl. &lt;/s&gt;&gt;)  Access statistics for D:/Thacsi/NNN/htk/gram/tri/trigram: lang model requested exact backed n/a mean stdev   bigram          0    0.0%  0.0%  0.0%  0.00  0.00   trigram         0    0.0%  0.0%  0.0%  0.00  0.00 </pre> <p>Kết quả lượng giá của <i>bigram</i></p> <pre> C:\Users\TuyenTran&gt;D:/Thacsi/NNN/htk/htk/LPlex.exe -n 2 -t D:/Thacsi/NNN/htk/gram/bi/bigram D:/Thacsi/NNN/htk/gram/ test_gram.txt LPlex test #0: 2-gram perplexity 1.#QNB, var -1.#IND, utterances 1, words predicted 0 num tokens 4, OOV 2, OOV rate 66.67% (excl. &lt;/s&gt;&gt;)  Access statistics for D:/Thacsi/NNN/htk/gram/bi/bigram: lang model requested exact backed n/a mean stdev   bigram          0    0.0%  0.0%  0.0%  0.00  0.00 </pre>
--	--

## 2.3 Các bước test:

<p><b>Bước</b></p> <p><b>1</b></p>	<ul style="list-style-type: none"> <li>- Chuẩn bị dữ liệu test: lấy tập dữ liệu test trong thư mục vivos.</li> <li>- Trong thư mục <b>test_1</b> và <b>test_2</b> dùng để lưu kết quả test.</li> <li>- Với mỗi bộ kết quả gồm 2 file: <i>recout.mlf</i> và <i>result.txt</i></li> </ul> <p><i>Cấu trúc test_1</i></p>  <p><i>Cấu trúc test_2</i></p> 
<p><b>Bước</b></p>	<ul style="list-style-type: none"> <li>- Tạo file <b>test.scp</b> và <b>test_mfc.scp</b>:</li> </ul>

2	<pre>python script/listwavmfc.py test/ test_mfc/ txt/test.scp txt/test_mfc.scp</pre> <ul style="list-style-type: none"><li>- Rút trích đặc trưng mfcc của các file test:</li></ul> <pre>htk/HCopy.exe -T 1 -C cfg/HCopy.cfg -S txt/test.scp</pre>										
Bước	<ul style="list-style-type: none"><li>- Tạo file <b>words.mlf</b> lưu trong thư mục <b>test_result</b>:</li></ul>										
3	<pre>perl script/prompts2mlf.pl test_result/words.mlf test_result/prompts.txt</pre>										
Bước	Thực hiện test dữ liệu:										
4	<pre>htk/HDecode.exe -H hmm/hmm15/macros -H hmm/hmm15/hmmdefs -S txt/test_mfc.scp -t 220.0 220.0 -C cfg/HDecode.cfg -i test_1/birecout.mlf -w gram/tri/trigram -p 0.0 -s 5.0 txt/dictHDecode.dct ins/tiedlist</pre> <p>Lưu ý:</p> <ul style="list-style-type: none"><li>- File từ điển <i>dictHDecode.dct</i> ở đây đã được thêm vào cuối file, 2 kí tự bắt đầu <b>&lt;s&gt;</b> và kết thúc <b>&lt;/s&gt;</b> của trigram</li></ul> <table border="1"><tr><td><b>&lt;s&gt;</b></td><td><b>[]</b></td><td><b>sil</b></td></tr><tr><td><b>&lt;/s&gt;</b></td><td><b>[]</b></td><td><b>sil</b></td></tr></table> <ul style="list-style-type: none"><li>- File HDecode.cfg thay đổi <b>STARTWORD = &lt;s&gt;</b> và <b>ENDWORD = &lt;/s&gt;</b></li></ul> <table border="1"><tr><td>TARGETKIND = MFCC_0_D_A</td></tr><tr><td>RAWMITFORMAT = T</td></tr><tr><td>STARTWORD = &lt;s&gt;</td></tr><tr><td>ENDWORD = &lt;/s&gt;</td></tr></table> <p>Sau khi thực hiện, tập tin recout.mlf có cấu trúc như sau:</p>	<b>&lt;s&gt;</b>	<b>[]</b>	<b>sil</b>	<b>&lt;/s&gt;</b>	<b>[]</b>	<b>sil</b>	TARGETKIND = MFCC_0_D_A	RAWMITFORMAT = T	STARTWORD = <s>	ENDWORD = </s>
<b>&lt;s&gt;</b>	<b>[]</b>	<b>sil</b>									
<b>&lt;/s&gt;</b>	<b>[]</b>	<b>sil</b>									
TARGETKIND = MFCC_0_D_A											
RAWMITFORMAT = T											
STARTWORD = <s>											
ENDWORD = </s>											

	 <pre> #!MLF!# "*/VIVOSDEV02_R106.lab" trowr neen thui ddooing . "*/VIVOSDEV02_R122.lab" cuxng khieesn cho hoj def dawit . "*/VIVOSDEV02_R130.lab" chij gawin hori anh theef soosng theef cheest laf khoong cos </pre>
<b>Bước</b>  <b>5</b>	<p>Thống kê kết quả, xuất kết quả ra tập tin txt.</p> <p><code>htk/HResults.exe -f -t -I test_1/words.mlf ins/tiedlist test_1/birecout.mlf &gt; test_1/biresult.txt</code></p> <p>Đây là kết quả của lần test dùng hmm15.</p>



```

biresult.txt
VIVOSDEV19_259.rec: 36.36( 36.36) [H= 4, D= 0, S= 7, I= 0, N= 11]
Aligned transcription: d:/Downloads/htk/test/test_mfc/VIVOSDEV19_259.lab vs d:/Downloads/
htk/test/test_mfc/VIVOSDEV19_259.rec
LAB: tisch cuwjc hown vovsi vaasn ddeef lieen quan ddeesn kieefu bafo
REC: tisch cuwjc hown tufy taafng ddeef lieeju vafng ddefn hieeru vafo
VIVOSDEV19_262.rec: 7.69( 7.69) [H= 1, D= 0, S= 12, I= 0, N= 13]
Aligned transcription: d:/Downloads/htk/test/test_mfc/VIVOSDEV19_262.lab vs d:/Downloads/
htk/test/test_mfc/VIVOSDEV19_262.rec
LAB: ddeer roofi chusng ta khao khast maxi trong vofng luaan hoofi nhaan gian
REC: chef roofi chufm safn tharo khasc maji traafm vojng luaajn hooji nhanh chef
VIVOSDEV19_266.rec: 9.09( 9.09) [H= 1, D= 0, S= 10, I= 0, N= 11]
Aligned transcription: d:/Downloads/htk/test/test_mfc/VIVOSDEV19_266.lab vs d:/Downloads/
htk/test/test_mfc/VIVOSDEV19_266.rec
LAB: vaasn ddeef nary sinh tuwf casi caau hori dinh cu aasy
REC: phaafn ddeef nafy xin tuwj taji caafu hoafnh thieejn thuj ngoafi
VIVOSDEV19_274.rec: 64.29( 64.29) [H= 9, D= 0, S= 5, I= 0, N= 14]
Aligned transcription: d:/Downloads/htk/test/test_mfc/VIVOSDEV19_274.lab vs d:/Downloads/
htk/test/test_mfc/VIVOSDEV19_274.rec
LAB: chij neen ddi khasm basc six vaf chojn phuowng phasp nguwa thai hieeju quar
REC: chij neen ddi khasm basc six veef choox phuowng pha nguowfi haji hieeju quar
VIVOSDEV19_279.rec: 27.27( 27.27) [H= 3, D= 0, S= 8, I= 0, N= 11]
Aligned transcription: d:/Downloads/htk/test/test_mfc/VIVOSDEV19_279.lab vs d:/Downloads/
htk/test/test_mfc/VIVOSDEV19_279.rec
LAB: nowj moojt cuojc chieesn choosng tham nhuxng vaaxn chuwa vafo ddaau
REC: nhau moojt cuojc dieexn chaajm phajm dufng vaajn duwa phari ddaau
----- Overall Results -----
SENT: %Correct=2.11 [H=16, S=744, N=760]
WORD: %Corr=41.48, Acc=40.08 [H=3203, D=109, S=4410, I=108, N=7722]
=====

```

Thực hiện kiểm thử trên bigram và trigram để có các kết quả khác.

### 3 Kịch bản thử nghiệm

- Tiến hành kiểm thử 1: qua 2 language model chạy trên toàn tập dữ liệu test trong thư mục *vivos/test* cho bigram.
- Tiến hành kiểm thử 2: qua 2 language model chạy trên 556 tập dữ liệu test trong thư mục *vivos/test*:
  - *Bigram*: chạy trên tập dữ liệu
  - *Trigram*: chạy trên tập dữ liệu

## 4 Kết quả, vẽ biểu đồ

### 4.1 Kết quả thực nghiệm

**Tiến hành kiểm thử 1:** với **bigram** và toàn bộ tập dữ liệu test.

Language Model	Độ chính xác %	Tổng số từ	Số từ đúng	Lỗi chèn
Bigram	40.08	7722	3203	108

**Tiến hành kiểm thử 2:** qua 2 language model: **bigram** và **trigram** và 556 dữ liệu test.

Language Model	Độ chính xác %	Tổng số từ	Số từ đúng	Lỗi chèn
Bigram	39.26	5138	2084	67
Trigram	41.94	5138	2225	70

### 4.2 Đánh giá kết quả

Qua kết quả thực nghiệm ở 2 lần thử nghiệm. Nhận thấy rằng:

- Xét trên tập dữ liệu đầy đủ, độ chính xác đạt được gần 40% với tổng số từ là 7722. Trong khi đó với tập dữ liệu 556 file thì độ chính xác là 39.26% với tổng số từ là 5138. Độ chính xác của việc thử nghiệm trên 1 phần bộ dữ liệu gần bằng tổng độ chính xác toàn tập dữ liệu.
- Xét trên hai mô hình bigram và trigram trên cùng tập dữ liệu thử nghiệm là 5138 từ, thì độ chính xác của trigram (41.94%) cao hơn bigram (39.26%). Tuy nhiên, thời gian để test dữ liệu khi dùng trigram, tốn nhiều thời gian hơn bigram.

Khi thực hiện bước build file gram từ cùng tập dữ liệu *train.txt* thì quá trình LBuild bigram chỉ tạo ra 2 file gram (gram.0, gram.1), còn với trigram thì tạo được 5 file gram (từ 72 gram.0 đến gram.4).

## 5 Đánh giá cá nhân

- Quá trình huấn luyện các hmm, tạo ra tiedlist với toàn bộ dữ liệu vivos tốn nhiều thời gian.
- Hdecode cho toàn bộ dữ liệu test, với mỗi file âm thanh tốn khoảng 30s để nhận dạng.
- Sử dụng trigram cho kết quả test cao hơn bigram nhưng tốn nhiều thời gian hơn.

## 6 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 tham khảo ở htk\_training\_vn.pdf và htkbook.pdf