

# WFST\_MT\_Small\_Corpus

July 19, 2025

## 1 WFST-based Machine Translation Tiny Demo

### 1.1 for Intern3 Students

by Ye Kyaw Thu, Lab Leader, LST Lab., Myanmar

Date: 18 July 2025

FST မောဒယ်တွေကို visualization လုပ်ဖို့ ရည်ရွယ်တာမှို့ ဗမာစာကြောင်း၊ ဆယ်ကြောင်း၊ ရခိုင်စာကြောင်း၊ ဆယ်ကြောင်း ကိုပဲ သုံးပြီး WFST basd machine translation demo လုပ်သွားပါမယ်။

### 1.2 Notes

WFST ကိုအခြေခံတဲ့ machine translation (MT) ကို လက်တွေ့လုပ်ပြတဲ့ notebook ပါ။

OpenFST နဲ့ MT evaluation အတွက် BLEU score, ChrF++ score တွက်တဲ့ ပရိုဂရမ်တွေကိုလည်း ကိုယ့် local machine ထဲမှာ ကြိုတင်ပြင်ဆင်ထားရပါလိမ့်မယ်။

Parallel corpus လည်း ကလည်း ဗမာ-ရခိုင်အတွက် ဆိုရင် ဗမာ-ရခိုင်၊ ဗမာ-ထားဝယ် အတွက် ဆိုရင်လည်း ဗမာ-ထားဝယ် ပြင်ဆင်ထားရပါလိမ့်မယ်။

Machine translation သုတေသနက တကယ်ကို ကျယ်ပြန်တာမှို့ အခု ဒီလက်တွေ့ သင်ခန်းစာမှာက WFST ကိုသုံးပြီး လုပ်တဲ့ MT အပိုင်းကိုပဲ အမိကထားပြောသွားပါမယ်။

### 1.3 Data Preparation

[40]: `cd /home/ye/exp/tiny_mt/alignment/`

`/home/ye/exp/tiny_mt/alignment`

[41]: `%%writefile train.my`  
ဘယ် အချိန် လဲ  
ဘာ ပြော သလဲ  
ဘာ လုပ် သလဲ  
မင်း ဘာ မေး မှာလဲ  
မင်း ဘာ ချက် လဲ  
အချိန် ရှိ လား  
ဘယ် အိမ် မှာ မင်း နေ သလဲ  
သူ ဘယ်မှာ နေ လဲ  
မင်း ဘာ အကြံပေး ချင် သလဲ

မင်း ဘာ တွေ စီမံ မှာလဲ

Overwriting train.my

[42]: !cat train.my

ဘယ် အချိန် လဲ  
ဘာ ပြော သလဲ  
ဘာ လုပ် သလဲ  
မင်း ဘာ မေး မှာလဲ  
မင်း ဘာ ချက် လဲ  
အချိန် ရှိ လား  
ဘယ် အိမ် မှာ မင်း နေ သလဲ  
သူ ဘယ်မှာ နေ လဲ  
မင်း ဘာ အကြံပေး ချင် သလဲ  
မင်း ဘာ တွေ စီမံ မှာလဲ

[43]: %%writefile train.rk

၁ ချိန် လေး  
၁ ပြော လေး  
၁ လုပ် လေး  
မင်း ၁ မိန်း ဖို့လေး  
မင်း ၁ ချက် လေး  
အချိန် ဟို လား  
၁ အိမ် မှာ မင်း နှီး လေး  
ယင်းသူ ၁မာ နှီး လေး  
မင်း ၁ အကြံပါး ချင် လေး  
မင်း ၁ တိ စီမံ ဖို့လေး

Overwriting train.rk

[44]: !cat train.rk

၁ ချိန် လေး  
၁ ပြော လေး  
၁ လုပ် လေး  
မင်း ၁ မိန်း ဖို့လေး  
မင်း ၁ ချက် လေး  
အချိန် ဟို လား  
၁ အိမ် မှာ မင်း နှီး လေး  
ယင်းသူ ၁မာ နှီး လေး  
မင်း ၁ အကြံပါး ချင် လေး  
မင်း ၁ တိ စီမံ ဖို့လေး

## 1.4 Alignment

MT အတွက် alignment ကလည်း အရေးကြီးတဲ့အပိုင်းပါ။ သုတေသနတွေ အများကြီး လုပ်ခဲ့ကြ၊ လုပ်နေကြပါတယ်။

ဒါ Lab exercise အတွက်က Anymalign ဆိုတဲ့ alignment tool ကို သုံးပါမယ်။

Link: <https://anymalign.limsi.fr/>

### 1.4.1 anymalign.sh

Alignment လုပ်ဖို့အတွက် သုံးခဲ့တဲ့ shell script က အောက်ပါအတိုင်းပါ။

[45]: !cat ./anymalign.sh

```
#!/bin/bash -v

SOURCE=$1;
TARGET=$2;

time python2.7 /home/ye/tool/anymalign/anymalign.py -i 5 -a 5 -w
./train.$SOURCE ./train.$TARGET > alignment-train.txt
wc ./alignment-train.txt

head ./alignment-train.txt
tail ./alignment-train.txt

cut -f1 ./alignment-train.txt > train-equal-smt.$SOURCE
cut -f2 ./alignment-train.txt > train-equal-smt.$TARGET

head ./train-equal-smt.$SOURCE
head ./train-equal-smt.$TARGET

tail ./train-equal-smt.$SOURCE
tail ./train-equal-smt.$TARGET

wc ./train-equal-smt.$SOURCE
wc ./train-equal-smt.$TARGET

echo "Alignment path:"
pwd;
```

## 1.5 Alignment

[46]: !wc train.my train.rk

```
10 40 448 train.my
10 40 467 train.rk
20 80 915 total
```

Commandline ကနေပဲ alignment လုပ်ပြမယ်  
-a 5 ၂ alignment ၂ ၅ per second ဆိုရင် ရပ်ပစ်လိုက်ဖို့  
-t 60 (60 sec ပြည့်တာနဲ့ ရပ်ခိုင်း)  
-w to get lexical weight

အသေးစိတ်က anymalign.py ၉ -help option နဲ့ run ပြီး လေ့လာပါ။

[47]: !time python2.7 /home/ye/tool/anymalign/anymalign.py --help

Usage: (basic usage)

```
python anymalign.py corpus.source corpus.target >translationTable.txt
```

For more control:

```
python anymalign.py [INPUT_FILE[.gz|.bz2] [...]] >ALIGNMENT_FILE  
python anymalign.py -m [ALIGNMENT_FILES[.gz|.bz2] [...]] >ALIGNMENT_FILE
```

INPUT\_FILE is a tab separated list of aligned sentences (1/line):  
<sentenceNlanguage1> [<TAB> <sentenceNlanguage2> [...]]

A generated ALIGNMENT\_FILE has the same format as INPUT\_FILE (same fields), plus three extra fields at the end of each line:

- 1) a space-separated list of lexical weights (1/language);
- 2) a space-separated list of translation probabilities (1/language);
- 3) an absolute frequency:

<phraseNlanguage1> [...] <TAB> <lexWeights> <TAB> <probas> <TAB> <frequency>

ALIGNMENT\_FILES is the concatenation of several ALIGNMENT\_FILE's.

Check out <http://users.info.unicaen.fr/~alardill/anymalign/> for more!

Options:

--version	show program's version number and exit
-h, --help	show this help message and exit
-m, --merge	Do not align. Input files are pre-generated alignment files (plain text format) to be merged into a single alignment file.
-T DIR, --temp-dir=DIR	(compatible with -m) Where to write temporary files. Default is OS dependant.
-q, --quiet	(compatible with -m) Do not show progress information on standard error.

Options to alter alignment behaviour:

-a NB_AL, --new-alignments=NB_AL	Stop alignment when number of new alignments per second is lower than NB_AL. Specify -1 to run indefinitely. [default: -1]
-i INDEX_N, --index-ngrams=INDEX_N	

Consider n-grams up to n=INDEX\_N as tokens. Increasing this value increases the number of long n-grams output, but slows the program down and requires more memory [default: 1]

-S NB\_SENT, --max-sentences=NB\_SENT  
Maximum number of sentences (i.e. input lines) to be loaded in memory at once. Specify 0 for all-in-memory. [default: 0]

-t NB\_SEC, --timeout=NB\_SEC  
Stop alignment after NB\_SEC seconds elapsed. Specify -1 to run indefinitely. [default: -1]

-w, --weight  
Compute lexical weights (requires additional computation time and memory).

#### Filtering options:

-D FIELDS, --discontiguous-fields=FIELDS  
Allow discontiguous sequences (like "give up" in "give it up") in languages at positions specified by FIELDS. FIELDS is a comma-separated list of integers (1-based), runs of fields can be specified by a dash (e.g. "1,3-5").

-l NB\_LANG, --min-languages=NB\_LANG  
Keep only those alignments that contain words in at least MIN\_LANGUAGES languages (i.e. columns). Default is to cover all languages.

-n MIN\_N, --min-ngram=MIN\_N  
Filter out any alignment that contains an N-gram with N < MIN\_N. [default: 1]

-N MAX\_N, --max-ngram=MAX\_N  
Filter out any alignment that contains an N-gram with N > MAX\_N (0 for no limit). [default: 7]

#### Output formatting options:

-d DELIM, --delimiter=DELIM  
Delimiter for discontiguous sequences. This can be any string. No delimiter is shown by default. Implies -D- (allow discontinuities in all languages) if -D option is not specified.

-e ENCODING, --input-encoding=ENCODING  
(compatible with -m) Input encoding. This is useful only for HTML and TMX output formats (see -o option). [default: utf-8]

-L LANG, --languages=LANG  
(compatible with -m) Input languages. LANG is a comma separated list of language identifiers (e.g. "en,fr,ar"). This is useful only for HTML (table headers) and TMX (<xml:lang>) output formats (see -o option).

```

-o FORMAT, --output-format=FORMAT
    (compatible with -m) Output format. Possible values
    are "plain", "moses", "html", and "tmx". [default:
    plain]

real    0m0.035s
user    0m0.026s
sys     0m0.009s

```

Manual တနောက် run ကြည့်ရအောင်...

[48]: !time python2.7 /home/ye/tool/anymalign/anymalign.py -i 5 -a 5 -w ./train.my ./train.rk > alignment-train.txt

```

Input corpus: 2 languages, 10 lines
Aligning... (ctrl-c to interrupt)
(13615 subcorpora, avg=3.16) Alignment done, proceeding...
Computing word cooccurrences...
Computing lexical weights...
108 alignments
Sorting alignments
Computing conditional probabilities...
Outputting results...
100%
real    0m2.039s
user    0m2.033s
sys     0m0.006s

```

Alignment လုပ်ပြီးထွက်လာတဲ့ output ဖိုင်ကို လေ့လာကြည့်ရအောင်

[49]: !wc ./alignment-train.txt

```
108 1083 10766 ./alignment-train.txt
```

Statistical Machine Translation (SMT) မှာ သုံးတဲ့ phrase table ပါပဲ။

[50]: !cat ./alignment-train.txt

မင်:	မင်:	1.000000	1.000000	0.885179	0.985707	48414				
မင်:	ဘာ	မင်:	၁၁	1.000000	0.750000	0.985405	0.883006	47398		
အချိန်	ရှိ	လား	အချိန်	ဟိ	လား	0.500000	1.000000	1.000000	0.956985	
35040										
ဘာ	လုပ်	သလဲ	၁၁	လုပ်	လေး	1.000000	0.428571	0.988822	1.000000	
32465										
ဘယ်	အချိန်	လဲ	၁၁	ချိန်	လေး	0.500000	0.107143	0.967509	1.000000	
32428										
ဘာ	ပြော	သလဲ	၁၁	ပြော	လေး	1.000000	0.428571	0.988264	1.000000	
32337										
သူ	ဘယ်မှာ	နေ	လဲ	ယင်းသူ	၁၁မာ	နီ	လေး	1.000000	0.428571	0.924981

1.000000	31096				
ဘယ် အိမ် မှာ မင်း နေ သလဲ		၈၁ အိမ် မှာ မင်း နီ လေး	1.000000	0.357143	
0.951807 1.000000	31027				
မင်း ဘာ တွေ စီမံ မှာလဲ မင်း ၉ တိ စီမံ ဖို့လေး	1.000000	0.750000		1.000000	
1.000000	30104				
မင်း ဘာ အကြံပေး ချင် သလဲ	မင်း ၈၁ အကြံပါး ချင် လေး		1.000000		
0.428571 0.993559 1.000000	29615				
မင်း ဘာ ချက် လဲ မင်း ၉၁ ချက် လေး	1.000000	0.428571		1.000000	
1.000000	29554				
မင်း ဘာ မေး မှာလဲ	မင်း ၈၁ မိန်း ဖို့လေး	1.000000	0.750000		1.000000
1.000000	29429				
ဘာ ၈၁ 1.000000 0.750000		1.000000 0.937768		25135	
နေ နီ လေး 1.000000 0.285714		0.841866 0.932468		13863	
အကြံပေး ချင် သလဲ	အကြံပါး ချင် လေး	1.000000	0.571429		0.872130
0.924956 12079					
ချက် လဲ ချက် လေး	1.000000 0.428571		0.847959 0.931635		11924
တွေ စီမံ မှာလဲ တိ စီမံ ဖို့လေး	1.000000 1.000000		0.986799 0.832171		
11811					
မေး မှာလဲ	မိန်း ဖို့လေး	1.000000 1.000000		0.985956 0.841432	
11584					
မှာလဲ ဖို့လေး 1.000000 1.000000		1.000000 1.000000		10796	
ဘာ ချက် လဲ ၈၁ ချက် လေး	1.000000 0.428571		0.827120 0.980738		
9674					
ဘာ တွေ စီမံ မှာလဲ	၈၁ တိ စီမံ ဖို့လေး	1.000000 0.750000		0.799899	
0.983678 9522					
ဘာ မေး မှာလဲ ၈၁ မိန်း ဖို့လေး		1.000000 0.750000		0.811355	
0.982730 9389					
ဘာ အကြံပေး ချင် သလဲ	၈၁ အကြံပါး ချင် လေး	1.000000 0.428571		0.	
903987					
0.962233 9274					
လဲ လေး 1.000000 0.428571		0.983079 0.549414		8250	
မင်း မင်း ၈၁ 1.000000 0.625000		0.114821 0.116994		6280	
သလဲ လေး 1.000000 0.571429		0.828669 0.404235		6070	
မင်း ဘာ တွေ စီမံ မင်း ၈၁ တိ စီမံ 1.000000 0.750000			1.000000		
1.000000 5398					
မင်း ဘာ မေး မင်း ၈၁ မိန်း 1.000000 0.750000			1.000000 1.000000		
5398					
လုပ် သလဲ	လုပ် လေး	1.000000 0.571429		0.736505 0.888404	
5294					
ပြော သလဲ	ပြော လေး	1.000000 0.571429		0.722829 0.876082	
5161					
ပြော ပြော 1.000000 1.000000		0.916599 0.550385		4506	
လုပ် လုပ် 1.000000 1.000000		0.920619 0.548054		4465	

	သူ ဘယ်မှာ နေ	ယင်းသူ အမာ နှီ	1.000000 1.000000	1.000000 0.618341
4086				
	ဓား ပိန်း	1.000000 1.000000	0.992485 1.000000	3962
	တွေ စီမံ	တိ စီမံ 1.000000 1.000000	0.992485 1.000000	3962
	ဘယ် အိမ် မှာ မင်း နေ	အိမ် မှာ မင်း နှီ	0.500000 1.000000	0.572954
0.686137	2737			
	နေ နှီ	1.000000 1.000000	0.158134 1.000000	2604
	သူ ဘယ်မှာ နေ လ ယင်းသူ အမာ နှီ	0.333333 1.000000	0.075019 0.381659	
2522				
	ဘာ တွေ စီမံ မှာလဲ	တိ စီမံ ဖို့လေး 0.333333 1.000000	0.200101	
0.167829	2382			
	ဘာ မေး မှာလဲ	မိန်း ဖို့လေး 0.333333 1.000000	0.188645 0.158568	
2183				
	ဘယ် အချိန်	၁ ချိန်	0.500000 0.250000	0.581842 0.857874
2179				
	အကြံပေး ချင်	အကြံပီး ချင်	1.000000 1.000000	1.000000 0.421576
2118				
	မင်း ဘာ ချက်	မင်း ၁ ချက်	1.000000 0.750000	1.000000 1.000000
1985				
	ပြော သလဲ	ပြော 0.250000 1.000000	0.277171 0.241725	1979
	ချက် လဲ ချက်	0.333333 1.000000	0.138529 0.393933	1948
	လုပ် သလဲ	လုပ် 0.250000 1.000000	0.263495 0.232478	1894
	ချက် ချက်	1.000000 1.000000	1.000000 0.359353	1777
	ဘာ လုပ် လုပ်	0.166667 1.000000	0.463097 0.196391	1600
	အကြံပေး ချင် သလဲ	အကြံပီး ချင် 0.250000 1.000000	0.114224	
0.314889	1582			
	ရှိ လား အချိန် ဟိ လား	1.000000 1.000000	1.000000 0.043015	1575
	ဘာ လုပ် ၁ လုပ် 1.000000 0.750000	0.455861 0.897948	1575	
	ဘယ် အိမ် မှာ မင်း နေ သလဲ	၁ အိမ် မှာ မင်း နှီ	1.000000 0.625000	
0.048193	0.457884	1571		
	ဘယ် အချိန် ချိန်	0.250000 1.000000	0.418158 0.333333	1566
	မင်း ဘာ အကြံပေး ချင်	မင်း ၁ အကြံပီး ချင်	1.000000 0.750000	1.000000
0.888502	1530			
	ဘယ် အိမ် မှာ မင်း နေ ၁	အိမ် မှာ မင်း နှီ	1.000000 0.625000	0.319238
0.444477	1525			
	ဘာ ပြော ပြော	0.166667 1.000000	0.458128 0.181752	1488
	ဘာ ပြော ၁ ပြော 1.000000 0.750000	0.443350 0.894410	1440	
	အချိန် လ ချိန်	0.166667 1.000000	0.724422 0.299915	1409
	သလဲ ၁ 1.000000 0.500000	0.171331 0.046823	1255	
	ဘာ အကြံပေး ချင် အကြံပီး ချင်	0.166667 1.000000	0.789157 0.234674	
1179				
	အိမ် မှာ မင်း နေ သလဲ	အိမ် မှာ မင်း နှီ	0.500000 1.000000	0.697509

0.294811	1176							
ဘာ ချက် လဲ	ချက်	0.055556	1.000000	0.098068	0.231951	1147		
သူ ဘယ်မှာ	ယင်းသူ	၁၉၆၁	1.000000	1.000000	1.000000	1.000000		
1004								
နေ လဲ နို့ လေး	1.000000	0.428571	1.000000	0.067532	1004			
ဘာ ချက် လဲ	ချက် လေး	0.666667	0.571429	0.074812	0.068365			
875								
ဘာ အကြံပေး ချင် သလဲ	အကြံပီး ချင် လေး	0.666667	0.571429	0.				
081879								
0.064323	840							
ဘယ် အချိန် လဲ	ချိန်	0.083333	1.000000	0.022436	0.160068	752		
မင်း ဘာ မင်း	0.666667	1.000000	0.014595	0.014293	702			
ဘယ် လေး	1.000000	0.285714	0.458590	0.044619	670			
အိမ် မှာ	အိမ် မှာ	1.000000	1.000000	1.000000	0.564929			
596								
ဘယ် အိမ် မှာ မင်း နေ	အိမ် မှာ မင်း နို့ လေး	1.000000	0.428571	0.107808				
0.746377	515							
ဘယ် အိမ် မှာ	အိမ် မှာ	0.500000	1.000000	1.000000	0.435071			
459								
ဘယ် ချိန်	ချိန်	0.500000	1.000000	0.284052	0.088335	415		
အချိန် ချိန်	0.500000	1.000000	0.867925	0.088123	414			
ပြော ပြော လေး	1.000000	0.142857	0.083401	0.069598	410			
လုပ် လုပ် လေး	1.000000	0.142857	0.079381	0.064608	385			
ဘယ် ၁၁	1.000000	0.250000	0.239562	0.013058	350			
ဘယ် အချိန် လဲ	ချိန် လေး	0.500000	0.428571	0.010055	0.658203			
337								
အချိန် လဲ	၁၁ ချိန်	0.333333	0.250000	0.172237	0.131890			
335								
အိမ် မှာ မင်း နေ သလဲ	၁၁ အိမ် မှာ မင်း နို့	1.000000	0.625000	0.198695				
0.097639	335							
ဘာ ပြော ပြော လေး	0.666667	0.571429	0.098522	0.054320	320			
ဘာ လုပ် လုပ် လေး	0.666667	0.571429	0.081042	0.046988	280			
ဘာ ပြော သလဲ ပြော	0.041667	1.000000	0.006540	0.026139	214			
မင်း ဘာ အကြံပေး ချင် သလဲ	မင်း ၁၁ အကြံပီး ချင်	1.000000	0.750000					
0.006441	0.111498	192						
ချက် လဲ ၁၁ ချက် လေး	1.000000	0.107143	0.013512	0.019262	190			
အကြံပေး ချင် သလဲ	၁၁ အကြံပီး ချင် လေး	1.000000	0.285714	0.013646				
0.019610	189							
ဘာ လုပ် သလဲ	လုပ်	0.041667	1.000000	0.005726	0.023076	188		
ဘာ လုပ် သလဲ	၁၁ လုပ်	1.000000	0.750000	0.005452	0.102052	179		
ဘာ အကြံပေး ချင် ၁၁ အကြံပီး ချင် လေး	1.000000	0.428571	0.117135					
0.018157	175							

အိမ် မှာ မင်း	နေ	သလဲ	အိမ် မှာ မင်း	နီ	လေး	1.000000	0.571429	0.103796
0.253623		175						
အချိန် လဲ		ချိန်	လေး			0.500000	0.428571	0.089974 0.341797
175								
ဘာ ပြော	သလဲ		ဘာ ပြော	1.000000	0.750000	0.005195	0.105590	170
မေး မှာလဲ			ဘာ မိန်း	ဖို့လေး		1.000000	0.250000	0.014044
0.017270		165						
တွေ စီမံ	မှာလဲ		ဘာ တိ	စီမံ	ဖို့လေး	1.000000	0.250000	0.013201
0.016322		158						
ဘာ တွေ	စီမံ		ဘာ တိ	စီမံ		1.000000	0.750000	1.000000 0.839572
157								
ဘာ မေး	ဘာ မိန်း		1.000000	0.750000		1.000000	0.839572	157
ဘာ အကြံပေး	ချင်	သလဲ	အကြံပါး	ချင်		0.041667	1.000000	0.014134
0.028861		145						
လဲ	ချိန်	0.333333	1.000000			0.016921	0.030226	142
ဘာ အကြံပေး	ချင်	အကြံပါး	ချင်	လေး		0.666667	0.571429	0.093708
0.010721		140						
အိမ် မှာ မင်း	နေ		အိမ် မှာ မင်း	နီ		1.000000	1.000000	1.000000
0.019052		76						
ဘယ် အိမ် မှာ မင်း			အိမ် မှာ မင်း	0.500000	1.000000		1.000000	
0.773196		75						
ဘာ ချက်	ချက်	0.166667	1.000000			1.000000	0.014762	73
အချိန် ဘာ		0.500000	0.125000			0.132075	0.002350	63
တွေ စီမံ			ဘာ တိ	စီမံ		1.000000	0.125000	0.007515 0.160428
30								
မေး	ဘာ မိန်း		1.000000	0.125000		0.007515	0.160428	30
ဘယ်	ဘာ ချိန်		1.000000	0.250000		0.017796	0.010236	26
အချိန် လဲ	လေး		0.500000	0.428571		0.013368	0.001731	26
အိမ် မှာ မင်း	အိမ် မှာ မင်း		1.000000	1.000000		1.000000	0.226804	
22								

shell script ထဲမှာ ရေးထားတဲ့ အလုပ်အကုန်ပြီးသွားတဲ့အခါမှာ အောက်ပါအတိုင်း ဖိုင်တွေရလိမ့်မယ်။

[51]: !pwd

```
/home/ye/exp/tiny_mt/alignment
```

[52]: !./anymalign.sh my rk

```
#!/bin/bash -v
```

```
SOURCE=$1;
TARGET=$2;
```

```
time python2.7 /home/ye/tool/anymalign/anymalign.py -i 5 -a 5 -w
```

```

./train.$SOURCE ./train.$TARGET > alignment-train.txt
Input corpus: 2 languages, 10 lines
Aligning... (ctrl-c to interrupt)
(13454 subcorpora, avg=3.17) Alignment done, proceeding...
Computing word cooccurrences...
Computing lexical weights...
108 alignments
Sorting alignments
Computing conditional probabilities...
Outputting results...
100%
real    0m2.035s
user    0m2.022s
sys     0m0.013s
wc ./alignment-train.txt
 108 1083 10766 ./alignment-train.txt

head ./alignment-train.txt
မင်း မင်း 1.000000 1.000000      0.885918 0.985898      48170
မင်း ဘာ မင်း ၈၁ 1.000000 0.750000      0.985423 0.882477      46578
အချိန် ရှိ လား အချိန် ဟိ လား 0.500000 1.000000      1.000000 0.957970
34918
ဘာ ပြော သလဲ ၈၁ ပြော လေး 1.000000 0.428571      0.988101 1.000000
31971
ဘာ လုပ် သလဲ ၈၁ လုပ် လေး 1.000000 0.428571      0.988000 1.000000
31697
ဘယ် အချိန် လဲ ၈၁ ချိန် လေး 0.500000 0.107143      0.967797 1.000000
31526
ဘယ် အိမ် မှာ မင်း နေ သလဲ ၈၁ အိမ် မှာ မင်း နှီ လေး 1.000000 0.357143
0.951705 1.000000      30525
သူ ဘယ်မှာ နေ လဲ ယင်းသူ အေမာ နှီ လေး 1.000000 0.428571      0.924504
1.000000      30504
မင်း ဘာ ချက် လဲ မင်း ၈၁ ချက် လေး 1.000000 0.428571      1.000000
1.000000      29182
မင်း ဘာ အကြံပေး ချင် သလဲ မင်း ၈၁ အကြံပါး ချင် လေး 1.000000
0.428571      0.993847 1.000000      29076
tail ./alignment-train.txt
အချိန် လဲ ချိန် လေး 0.500000 0.428571      0.055157 0.245098
100
ဘာ ချက် ချက် 0.166667 1.000000      1.000000 0.018051      90
ဘယ် အိမ် မှာ မင်း အိမ် မှာ မင်း 0.500000 1.000000      1.000000
0.747664      80
အိမ် မှာ မင်း နေ အိမ် မှာ မင်း နှီ 1.000000 1.000000      1.000000
0.019061      69

```

အချိန်	၈၁	0.500000	0.125000	0.103672	0.001771	48			
မေး	၈၁	မိန္ဒာ		1.000000	0.125000	0.011892	0.265896	46	
တွေ	စီမံ	၈၁	တိ	စီမံ	1.000000	0.125000	0.011892	0.265896	
46									
အိမ်	မှာ	မင်း	အိမ်	မှာ	မင်း	1.000000	1.000000	1.000000	0.252336
27									
ဘယ်	၈၁	ချိန်		1.000000	0.250000	0.015613	0.008663	21	
အချိန်	လဲ	လေး		0.500000	0.428571	0.011583	0.001370	21	

```
cut -f1 ./alignment-train.txt > train-equal-smt.$SOURCE
```

```
cut -f2 ./alignment-train.txt > train-equal-smt.$TARGET
```

```
head ./train-equal-smt.$SOURCE
```

မင်း

မင်း ဘာ

အချိန် ရှိ လား

ဘာ ပြော သလဲ

ဘာ လှပ် သလဲ

ဘယ် အချိန် လဲ

ဘယ် အိမ် မှာ မင်း နေ သလဲ

သူ ဘယ်မှာ နေ လဲ

မင်း ဘာ ချက် လဲ

မင်း ဘာ အကြံပေး ချင် သလဲ

```
head ./train-equal-smt.$TARGET
```

မင်း

မင်း ၈၁

အချိန် ဟို လား

၈၁ ပြော လေး

၈၁ လှပ် လေး

၈၁ ချိန် လေး

၈၁ အိမ် မှာ မင်း နှီး လေး

ယင်းသူ ဘမာ နှီး လေး

မင်း ၈၁ ချက် လေး

မင်း ၈၁ အကြံပါး ချင် လေး

```
tail ./train-equal-smt.$SOURCE
```

အချိန် လဲ

ဘာ ချက်

ဘယ် အိမ် မှာ မင်း

အိမ် မှာ မင်း နေ

အချိန်

မေး

```

တွေ့ စီမံ
အိမ် မှာ မင်း
ဘယ်
အချိန် လဲ
tail ./train-equal-smt.$TARGET
ချိန် လေး
ချက်
အိမ် မှာ မင်း
အိမ် မှာ မင်း နဲ့
၁၁
၁၁ မိန်း
၁၁ တိ စီမံ
အိမ် မှာ မင်း
၁၁ ချိန်
လေး

wc ./train-equal-smt.$SOURCE
108 283 3229 ./train-equal-smt.my
wc ./train-equal-smt.$TARGET
108 260 3143 ./train-equal-smt.rk

echo "Alignment path:"
Alignment path:
pwd;
/home/ye/exp/tiny_mt/alignment

```

[75]: !ls \*

```

alignment-train.txt  test.my          train-equal-smt.my  train.my
anyalign.sh          train-equal-smt. train-equal-smt.rk  train.rk

```

[76]: !wc train.{my,rk}

```

10 40 448 train.my
10 40 467 train.rk
20 80 915 total

```

## 1.6 Test Data Preparation

test data ကို စာကြောင်း တစ်ကြောင်းတည်းနဲ့ပဲ ပြင်ဆင်ပါမယ်။

[53]: %%writefile test.my  
ဘယ် အချိန် ဘာ လုပ် သလဲ

Overwriting test.my

ရခိုင် ဘာသာစကားအတွက် test စာကြောင်းကိုလည်း ပြင်မယ်။

[54] : `%%writefile test.rk`  
၈၁ ချိန် ၈၁ လုပ် လေး

Overwriting test.rk

[55] : `!wc test.{my,rk}`

```
1    5  59 test.my  
1    5  53 test.rk  
2 10 112 total
```

## 1.7 Preparation for MT Folder

MT ကို my-rk ဖိုလ်ဒါထဲမှာ run ပါမယ်။

WFST MT လုပ်ဖို့အတွက် ကြိုရေးထားတဲ့ perl, python, shell script တွေကို အဲဒီ my-rk ဖိုလ်ဒါအောက်ထဲကို ကော်ပိုကူးယူပါမယ်။

အထက်က alignment လုပ်ပြီး ထွက်လာတဲ့ ဖိုင်ကနေ source column, target column နှစ်ခုကိုပဲဖြတ်ယူပါမယ်။

[81] : `!pwd`

```
/home/ye/exp/tiny_mt/alignment
```

[57] : `cd ../`

```
/home/ye/exp/tiny_mt
```

[86] : `mkdir my-rk`

[87] : `!ls`

```
alignment  my-rk
```

[88] : `!cp /home/ye/exp/wfst_mt/my-rk/*.* ./my-rk/`

[59] : `!ls ./my-rk/*sh`

```
./my-rk/eval.sh          ./my-rk/test-nofstdraw.sh  
./my-rk/mk-train-symbol.sh   ./my-rk/test.sh  
./my-rk/mk-uniq-word.sh     ./my-rk/train-test-eval.sh  
./my-rk/multi-test.sh      ./my-rk/translate-nofstdraw.sh  
./my-rk/shortest-path-to-line.sh ./my-rk/translate.sh
```

[90] : `!cp /home/ye/exp/wfst_mt/my-rk/*.*py ./my-rk/`

[60] : `!ls ./my-rk/*.*py`

```
./my-rk/align_ibm.py       ./my-rk/mk_fst_format.py  
./my-rk/extract_column_symbols.py ./my-rk/mk-symbol.py  
./my-rk/make_ngram_fst.py
```

ရှောက run ခဲ့တဲ့ anymalign.sh နဲ့ ဖြတ်ထားပြီးသားမို့လို့ filename ပြောင်းပြီး ကော်ပီကူးယူပါမယ်

[94]: !cp ./alignment/train-equal-smt.my ./my-rk/all.my

[95]: !cp ./alignment/train-equal-smt.rk ./my-rk/all.rk

မြင်သာအောင် parallel data or aligned data ကို head command နဲ့ ကြည့်ရအောင်။

[61]: !head ./my-rk/all.my

မင်း  
မင်း ဘာ  
အချိန် ရှိ လား  
ဘာ ပြော သလဲ  
ဘာ လုပ် သလဲ  
ဘယ် အချိန် လဲ  
သူ ဘယ်မှာ နေ လဲ  
ဘယ် အိမ် မှာ မင်း နေ သလဲ  
မင်း ဘာ တွေ စီမံ မှာလဲ  
မင်း ဘာ ချက် လဲ

[62]: !head ./my-rk/all.rk

မင်း  
မင်း ဘာ  
အချိန် ဟို လား  
ဘာ ပြော လေး  
ဘာ လုပ် လေး  
ဘာ ချိန် လေး  
ယင်းသူ ဘမာ နှီး လေး  
ဘာ အိမ် မှာ မင်း နှီး လေး  
မင်း ဘာ တိ စီမံ ဖို့လေး  
မင်း ဘာ ချက် လေး

## 1.8 Move to MT Folder and Check Programs

Machine Translation စမ်းသပ်မှုလုပ်မယ့် folder အောက်တဲ့မှာ alignment လုပ်ထားတဲ့ parallel data တွေအပြင် WFST-based MT လုပ်ဖို့အတွက် ပရိုဂရမ်တချို့ကို ကြိုပြင်ဆင်ထားရပါမယ်။ (အထက်မှာ ပြထားတဲ့ ဥပမာ အတိုင်း)

အခိုက် OpenFST command တွေကိုတော့ bash shell script ထဲမှာပဲရေးထားပြီး run ပါတယ်။

[98]: cd ./my-rk

/home/ye/exp/tiny\_mt/my-rk

## 1.9 Note

တကယ် corpus အကြီးနဲ့ WFST MT ကို run တဲ့အခါမှာ open test data ၏ parallel corpus ထဲကနေ သပ်သပ်ထဲတော်မူတဲ့ ရတာပါ။ အဲဒါကြာင့် symbol ဖိုင်ဆောက်တဲ့အခါမှာ unknown word တွေကို သီးသန့်ကိုင်တွယ်တာမလုပ်ချင်လို့ training + test data ကို ပေါင်းပြီး all.my (i.e. source), all.rk (i.e. target) အဖြစ် ဖိုင်အသစ်တွေ ဆောက်ပါတယ်။ သို့သော် ဒီ small corpus နဲ့ run တဲ့ ဒီမိမှာတော့ အဲဒီလိုတွေ သီးသန့်ပြင်မနေတော့ပဲ train-equal-smt.my, train-equal-smt.rk ကနေပဲ ကူးယူလိုက်တာပါ။ WFST MT အတွက် ရေးထားတဲ့ shell script မှာတော့ ခွဲသုံးတာကို တွေ့ရပါလိမ့်မယ်။

```
[64]: cd /home/ye/exp/tiny_mt/my-rk/
```

```
/home/ye/exp/tiny_mt/my-rk
```

```
[102]: !cp ./alignment/train-equal-smt.* .
```

## 1.10 Test Data

Alignment folder အောက်မှာ ပြင်ခဲ့တဲ့ test data ဖိုင်ကိုလည်း my-rk/ အောက်ကို ကော်ပီကူးယူမယ်။

```
[105]: !cp ./alignment/test.* .
```

## 1.11 train-test-eval.sh

Alignment လုပ်ထားတဲ့ parallel corpus နဲ့ ဒီ train-test-eval.sh shell script ကို run လိုက်ရှုံးနဲ့ training, testing, evaluation အကုန်လုပ်ပေးသွားပါလိမ့်မယ်။

```
#!/bin/bash
```

```
# Prepare oneline test data
# head ၏ ပုံစံမ တစ်ခုတည်းဖြစ်နေလို့ tail ကို သုံးဖို့ ဆုံးဖြတ်လိုက်တယ်
tail -n 1 ./train-equal-smt.my > oneline.my

# Building Transducers with training data (i.e. language model, translation model, composing etc)
time ./translate-nofstdraw.sh ./train-equal-smt.my ./train-equal-smt.rk oneline.my ./all.my ./all.rk

# Testing with WFST MT
time ./multi-test.sh ./all.my ./all.rk ./test.my 2>&1 | tee anymaTrainingDataOnly-test-myrk.log

# Evaluation
time ./eval.sh ./test.rk hyp.txt.clean
```

## 1.12 translate-nofstdraw.sh

nofstdraw ဆိတာက OpenFST framework ရဲ့ command တရာဖြစ်တဲ့ fstdraw နဲ့ graph ပုံတွေ မထုတ်ပဲ run ခိုင်းတာပါ။ ဒေတာက များရင် PDF သို့မဟုတ် png ဖိုင်နဲ့ ကြည့်ရင် အဆင်မပြောလိုပါ။

```
[65]: cd /home/ye/exp/tiny_mt/my-rk/
```

```
/home/ye/exp/tiny_mt/my-rk
```

./translate-nofstdraw.sh က အောက်ပါအတိုင်းပါ။ တကယ့် experiment လုပ်တုန်းက သုံးခဲ့တာမို့ တရာ့၏ comment တွေနဲ့ရှုပ်နေတာတွေ ရှိပါတယ်။ ဒါတောင် တော်တော်လေး ဖျက်စရာရှိတာတွေ ဖျက်ထားတဲ့ ဗားရှင်းပါ။

```

#!/bin/bash
set -e

# written by Ye Kyaw Thu, LU Lab., Myanmar
# before running this shell script, you have to run "mk-train-symbol.sh" for both source and target language
# Build FST Translation Model and test with one line example:
# $ ./translate-nofstdraw.sh ./train.my ./train.ro ./oneline.my ./all.my ./all.ro

corpuse=$1;
corpusf=$2;
input=$3;

# open testing လုပ်တဲ့အခါမှာ symbol တွေက မရှိရင် error တက်လို့ ...
allsource=$4;
alltarget=$5;

# Create symbol file for target language
echo "Create symbol file for target language ...";
./mk-uniq-word.sh $corpusf > $corpusf.words
#perl ./mk-symbol.pl ./$corpusf.words > $corpusf.words.sym
python ./mk-symbol.py --input $corpusf.words --output $corpusf.words.sym

# Create symbol file
#echo "Create symbol file for source language ...";
#./mk-uniq-word.sh $corpuse > $corpuse.words
#python ./mk-symbol.py --input $corpusf.words --output $corpusf.words.sym

# Prepare test data FST
#perl ./mk-fst-format.pl $input > $input.formatted
python ./mk_fst_format.py --input $input --output $input.formatted
inputfst=$input.formatted
echo "Preparing test data FST finished!"
echo "Test sentence:"; cat $input;

# Create a bigram language model from the corpus
python make_ngram_fst.py < $corpusf --n 2 > bigram.txt
python extract_column_symbols.py --column 2 < bigram.txt > bigram.isym
fstcompile --keep_isymbols --keep_osymbols --isymbols=bigram.isym --osymbols=bigram.isym bigram
echo "fstcompile for the bigram language model FST finished!"
#fstdraw --portrait --acceptor --show_weight_one --ssymbols=$corpuse.words.sym bigram.fst bigram
#dot -Tps:cairo bigram.dot > bigram.ps
#ps2pdf bigram.ps
#pdfcrop bigram.pdf
#mv bigram-crop.pdf bigram.pdf
#evince ./bigram.pdf

```

```

# for translation model
python align_ibm.py --source $corpus --target $corpusf --output onetoone.txt
# try to cover open test symbol ...
python align_ibm.py --source $allsource --target $alltarget --output all_src_trg.txt
python extract_column_symbols.py --column 2 < all_src_trg.txt > onetoone.isym
python extract_column_symbols.py --column 3 < all_src_trg.txt > onetoone.osym

fstcompile --keep_isymbols --keep_osymbols --isymbols=onetoone.isym --osymbols=onetoone.osym onetoone.txt
echo "fstcompile for the translation model FST finished!"
fstdraw --portrait --acceptor --show_weight_one onetoone.fst onetoone.dot
#dot -Tps:cairo onetoone.dot > onetoone.ps
#ps2pdf onetoone.ps
#pdfcrop onetoone.pdf
#mv onetoone-crop.pdf onetoone.pdf
#evince onetoone.pdf

# Compose together a translation model and language model
fstcompile --keep_isymbols --keep_osymbols --isymbols=train.my.words.sym --osymbols=bigram.isym onetoone.fst
fstcompose onetoone.fst bigram.fst composed.fst
fstcompile --keep_isymbols --keep_osymbols --isymbols=onetoone.isym --osymbols=bigram.isym onetoone.fst
echo "compile success!!";

fstcompose onetoone.fst bigram.fst composed.fst
echo "fstcompose together a translation model and language model finished!"
fstdraw --portrait --show_weight_one composed.fst composed.dot
#dot -Tps:cairo composed.dot > composed.ps
#ps2pdf composed.ps
#pdfcrop composed.pdf
#mv composed-crop.pdf composed.pdf
#evince composed.pdf

# Formulate the input as a WFST
fstcompile --keep_isymbols --keep_osymbols --isymbols=onetoone.isym --osymbols=onetoone.isym ${inputfst%.*}.txt
echo "fstcompile for the input sentence finished!"
fstdraw --portrait --acceptor ${inputfst%.*}.fst ${inputfst%.*}.dot
#dot -Tps:cairo ${inputfst%.*}.dot > ${inputfst%.*}.ps
#ps2pdf ${inputfst%.*}.ps
#pdfcrop ${inputfst%.*}.pdf
#mv ${inputfst%.*}-crop.pdf ${inputfst%.*}.pdf
#evince ${inputfst%.*}.pdf

# Compose together into a search graph
fstcompose ${inputfst%.*}.fst composed.fst search.fst
echo "fstcompose together into a search graph finished!"
fstdraw --portrait search.fst search.dot
#dot -Tps:cairo search.dot > search.ps

```

```

#ps2pdf search.ps
#pdfcrop search.pdf
#mv search-crop.pdf search.pdf
#evince ./search.pdf

# Remove epsilon to make it easier to read
fstrmepson search.fst searchrmeps.fst
echo "fstrmepson finished!"
fstdraw --portrait searchrmeps.fst searchrmeps.dot
dot -Tps:cairo searchrmeps.dot > searchrmeps.ps
ps2pdf searchrmeps.ps
pdfcrop searchrmeps.pdf
mv searchrmeps-crop.pdf searchrmeps.pdf
#evince ./searchrmeps.pdf

# Print the shortest path
fstshortestpath ./searchrmeps.fst > shortest-path.fst
echo "finding the shortest path finished!"
fstdraw --portrait --isymbols=onetoone.isym --osymbols=$corpusf.words.sym ./shortest-path.fst

#evince ./shortest-path.pdf

# Shortest-path to normal sentence
bash ./shortest-path-to-line.sh ./shortest-path.fst

train-test-eval.sh ကို run ထာက corpus ရဲ့ ပမာဏပေါ် မှတည်ပြီး ကြာပါလိမ့်မယ်။ ဒီတခါ ကတော့
စာကြောင်း သောကြောင်း parallel corpus ဖို့လို့ မြန်ပါလိမ့်မယ်...

```

[68]: ./train-test-eval.sh

```

Create symbol file for target language ...
Preparing test data FST finished!
Test sentence:
အချိန် လဲ
fstcompile for the bigram language model FST finished!
fstcompile for the translation model FST finished!
compile success!!
fstcompose together a translation model and language model finished!
fstcompile for the input sentence finished!
fstcompose together into a search graph finished!
fstrmepson finished!
finding the shortest path finished!
၅၀ ချိန် </s>
real    0m0.257s
user    0m0.196s
sys     0m0.082s
mv hyp.txt hyp.old
Translation: ဘယ် အချိန် ဘာ လုပ် သလဲ

```

```

hypothesis file: hyp.txt.clean

real      0m0.084s
user      0m0.062s
sys       0m0.027s
Evaluation with BLEU score:
BLEU = 100.00, 100.0/100.0/100.0/100.0 (BP=1.000, ratio=1.000, hyp_len=5,
ref_len=5)
Evaluation with chrF++ score:
start_time:      1752815850
c6+w2-F2        100.0000
c6+w2-avgF2     100.0000
end_time:       1752815850

real      0m0.024s
user      0m0.017s
sys       0m0.007s

```

အထက်မှာ မြင်ရတဲ့အတိုင်းပါပဲ BLEU score က 100.00 ရပါတယ်။ Known word တွေနဲ့ပဲ ပြီးတော့ တစ်ကြောင်းထဲကို ဘာသာပြန်ခြင်းတာ မိမ့်လိုပါ။ လက်တွေ့ machine translation လုပ်တဲ့အခါမှာတော့ ဒီလိမ့်မျိုး score က မဖြစ်နိုင်ပါဘူး။

### 1.13 Let's Run Step by Step

အထက်မှာ run ပြခဲ့တာက Transducer model ဆောက်တာကနေ test data နဲ့ translate လုပ်ပြီး evaluation လုပ်သွားတဲ့ အဆင့်အကုန်လုံးပါပဲ။ အဲဒါကြောင့် လက်ရှိ ပြန့် ဒီမိုအတွက် WFST based translation pipeline တစ်ခုလုံးက အဆင်ပြေတယ်၊ ဘာ error မှ မရှိနိုင်တော့ဘူး ဆိုတာ သေချာသွားပါပြီ။ ခုချိန်က စပြီး အရေးကြီးတဲ့ အဆင့် တစ်ဆင့်ချင်းစီကို command ပေး run သွားရင်းနဲ့ FST တွေကို graph အနေနဲ့ ထုတ်ကြည့်ရင်း လေ့လာသွားကြရအောင်။

### 1.14 Create Symbol File For Target Language

```
[69]: !pwd
/home/ye/exp/tiny_mt/my-rk

[73]: # Target language က ရှိုင်ဘာသာစကားမို့လို့ train-equal-smt.rk ဖိုင်ကို သုံးပါမယ်
!echo "Create symbol file for target language ...";
!./mk-uniq-word.sh train-equal-smt.rk > train-equal-smt.rk.words
#!/perl ./mk-symbol.pl ./train-equal-smt.rk.words > train-equal-smt.rk.words.sym
!python ./mk-symbol.py --input ./train-equal-smt.rk.words --output
↳ train-equal-smt.rk.words.sym
```

Create symbol file for target language ...

Target language အတွက် ရလာတဲ့ symbol ဖိုင်ကို ရိုက်ထုတ်ကြည့်ရအောင်။

```
[74]: !cat train-equal-smt.rk.words.sym
```

```

<s> 0
NULL 1
ချက် 2
ချင် 3
ချိန် 4
စီမံ 5
၁၁ 6
၁၁၁၁ 7
တိ 8
နှု 9
ငြာ 10
ဖိုလေး 11
မင်း 12
မိန်း 13
မှာ 14
ယင်းသူ 15
လား 16
လုပ် 17
လေး 18
ဟိ 19
အကြံး 20
အချိန် 21
အိမ် 22
</s> 23

```

## 1.15 Create Symbol File for Source Language

```
[75]: # Create symbol file
#echo "Create symbol file for source language ...";
!./mk-uniq-word.sh train-equal-smt.my > train-equal-smt.my.words
#!perl ./mk-symbol.pl ./train-equal-smt.my.words > train-equal-smt.my.words.sym
!python ./mk-symbol.py --input ./train-equal-smt.my.words --output
 ↵train-equal-smt.my.words.sym
```

Source အတွက် ပြင်ဆင်ထားတဲ့ symbol ဖိုင်ကို ရှိက်ထုတ်ကြည့်ရအောင်။

```
[76]: !cat ./train-equal-smt.my.words.sym
```

```

<s> 0
NULL 1
ချက် 2
ချင် 3
စီမံ 4
၁၂ 5
နေ့ 6

```

ପ୍ରେସ୍ 7  
ବାଯିଁ 8  
ବାଯିଁମୁକ୍ତି 9  
ବାବୀ 10  
ମଣିଃ 11  
ମେ 12  
ମୁକ୍ତି 13  
ମୁକ୍ତିଲ୍ 14  
ମୁଖୀ 15  
ଲବ୍ଧି 16  
ଲୁହି 17  
ଲେ 18  
ଲବ୍ଧି 19  
ଲୁହି 20  
ଆଗ୍ରହିତିକାରୀ 21  
ଆଶ୍ଵିନ୍ଦିକାରୀ 22  
ଆଶ୍ଵିନ୍ଦିକାରୀ 23  
</s> 24

## 1.16 Notes

လက်တွေ WFST နဲ့ test ဖိုင်တစ်ဖိုင်လုံးကို ဘာသာပြန်တဲ့အခါ စာကြောင်းရေ ရှိရင် ရှိသလောက် အချိန်ကြာပါတယ်။ အဲဒါကြောင့် အရင်ဆုံး oneline.my ဆိတ် စာကြောင်း တစ်ကြောင်းထဲ closed data ကို ပြင်ပြီးမှ စမ်းကြည့်ပါတယ်။ အဲဒီအဆင့်မှာတင် fail ဖြစ်သွားရင် ပြင်စရာရှိတာ ပြင်နိုင်အောင်လိုပါ။ သို့သော်လည်း လက်ရှိ ဒီမိုပြန့် ပြင်ထားတဲ့ test data ကလည်း graph ထုတ်ပြီးကြည့်ရင် အရမ်းမများအောင်လို့ တကယ်တမ်းမှာတော့ တစ်ကြောင်းတည်းနဲ့ပဲ translation လုပ်မှာပါပဲ။

## 1.17 multi-test.sh

multi-test.sh ဖိုင်က အောက်ပါအတိုင်းပါ။

```
#!/bin/bash
```

```
# written by Ye Kyaw Thu, LU Lab., Myanmar
# e.g. $ ./multi-test.sh all.my all.ro head.my
#
source=$1;
target=$2;
testdata=$3;

# Backup original hyp file
echo "mv hyp.txt hyp.old";
mv hyp.txt hyp.old

cat $testdata | while read -r line
do
```

## 1.18 Build Bigram LM

```
[83]: # Create a bigram language model from the corpus
!python make_ngram_fst.py --input all.rk --output bigram.txt --n 2
!python extract_column_symbols.py --column 2 --input bigram.txt --output bigram.
    ↪isym

!fstcompile --keep_isymbols --keep_osymbols --isymbols=bigram.isym
    ↪--osymbols=bigram.osym bigram.txt bigram.fst
!echo "fstcompile for the bigram language model FST finished!"
```

fstcompile for the bigram language model FST finished!

```
[84]: cd /home/ye/exp/tiny_mt/my-rk/
```

```
[85]: !python make_ngram_fst.py < all.rk > bigram.txt  
!python extract_column_symbols.py --column 2 --input bigram.txt --output bigram.  
    ↵isym
```

```
!fstcompile --keep_isymbols --keep_osymbols --isymbols=bigram.isym  
  ↵--osymbols=bigram.isym bigram.txt bigram.fst  
!echo "fstcompile for the bigram language model FST finished!"
```

fstcompile for the bigram language model FST finished!

[86]: !fstdraw --portrait --isymbols=./bigram.isym --osymbols=./bigram.isym ./bigram.  
 ↵fst | dot -Tpdf -Gmargin=0 > ./bigram.pdf

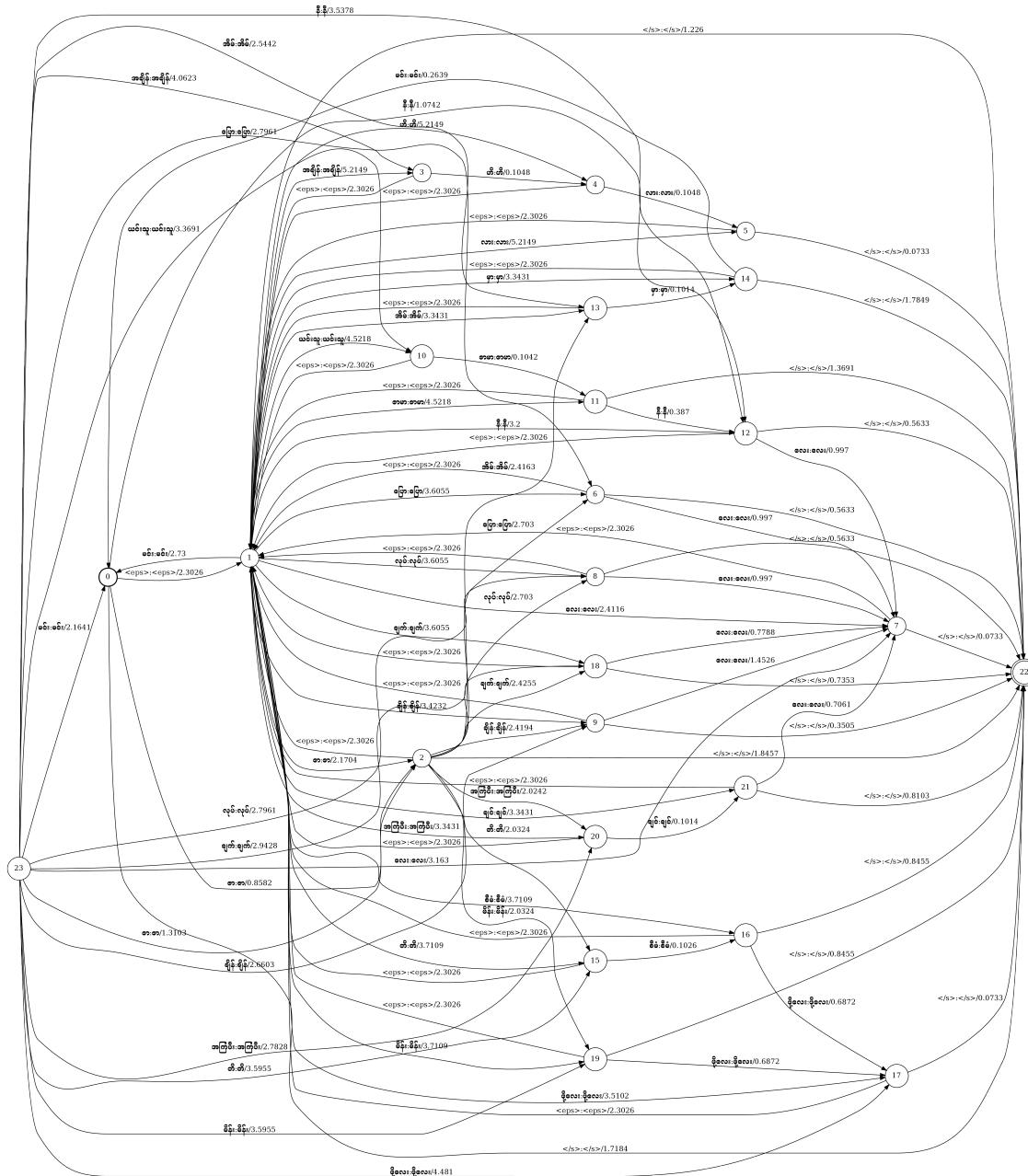
[87]: !pdffcrop bigram.pdf bigram\_cropped.pdf

PDFCROP 1.42, 2023/04/15 - Copyright (c) 2002-2023 by Heiko Oberdiek, Oberdiek  
Package Support Group.  
==> 1 page written on `bigram\_cropped.pdf'.

[88]: !convert -density 800 bigram\_cropped.pdf -quality 100 bigram\_cropped.png

[89]: from IPython.display import Image  
Image(filename="bigram\_cropped.png")

[89]:



## 1.19 Build IBM Model-1 or One-to-One Translation FST

 ...

[90]: !python align\_ibm.py --help

```
usage: align_ibm.py [-h] --source SOURCE --target TARGET [--output OUTPUT]
                   [--model {1}] [--version]
```

Train IBM Model 1 aligner and output in FST format.

```
options:
    -h, --help                  show this help message and exit
    --source SOURCE, -s SOURCE
                                Source language file
    --target TARGET, -t TARGET
                                Target language file
    --output OUTPUT, -o OUTPUT
                                Output file for FST format (default: stdout)
    --model {1}, -m {1}          IBM model version (only Model 1 supported)
    --version                   show program's version number and exit
```

IBM Model-1 က မြန်မာ-ရခိုင်လို spoken dialogue အတွက် အတိုင်းအတာ တစ်ခုထိ အလုပ်လုပ်ပေးပါတယ်။

"""

*IBM Alignment Model-1  
Written by Ye Kyaw Thu.  
Last updated: 18 July 2025*

"""

```
import math
import argparse
from collections import defaultdict
from typing import List, Tuple, TextIO

def read_parallel_corpus(src_file: TextIO, tgt_file: TextIO) -> Tuple[List[List[List[str]]], List[List[List[str]]]]:
    src_lines = [line.strip().split() for line in src_file]
    tgt_lines = [line.strip().split() for line in tgt_file]
    if len(src_lines) != len(tgt_lines):
        raise ValueError(f"Line count mismatch: source={len(src_lines)} target={len(tgt_lines)}")
    return src_lines, tgt_lines

def train_ibm_model_1(src_sents: List[List[str]], tgt_sents: List[List[str]]) -> Tuple[defaultdict(int), defaultdict(int)]:
    joint_counts = defaultdict(int)
    tgt_counts = defaultdict(int)

    for src, tgt in zip(src_sents, tgt_sents):
        for f in src:
            for e in tgt:
                joint_counts[f, e] += 1
                tgt_counts[e] += 1

    return joint_counts, tgt_counts
```

```

def write_fst_output(joint_counts: defaultdict, tgt_counts: defaultdict, output: TextIO):
    for (f, e), count in joint_counts.items():
        prob = count / tgt_counts[e]
        logprob = 0 if prob == 1 else -math.log(prob)
        print(f"0 0 {f} {e} {logprob:.4f}", file=output)
    print("0 0 </s> </s> 0", file=output)
    print("0", file=output)

def main():
    parser = argparse.ArgumentParser(
        description="Train IBM Model 1 aligner and output in FST format."
    )
    parser.add_argument('--source', '-s', type=argparse.FileType('r'), required=True,
                        help="Source language file")
    parser.add_argument('--target', '-t', type=argparse.FileType('r'), required=True,
                        help="Target language file")
    parser.add_argument('--output', '-o', type=argparse.FileType('w'), default='-', 
                        help="Output file for FST format (default: stdout)")
    parser.add_argument('--model', '-m', choices=['1'], default='1',
                        help="IBM model version (only Model 1 supported)")
    parser.add_argument('--version', action='version', version='IBMAccelerator 1.0')

    args = parser.parse_args()

    try:
        src_sents, tgt_sents = read_parallel_corpus(args.source, args.target)
    except ValueError as e:
        print(f"[ERROR] {e}")
        exit(1)

    joint_counts, tgt_counts = train_ibm_model_1(src_sents, tgt_sents)
    write_fst_output(joint_counts, tgt_counts, args.output)

if __name__ == '__main__':
    main()

```

[91]: !python align\_ibm.py --source all.my --target all.rk --output onetoone.txt

[92]: !cat onetoone.txt

```

0 0 ພົມ: ພົມ: 1.3863
0 0 ພົມ: ເວ 2.1203
0 0 ວິວ ພົມ: 2.1665
0 0 ວິວ ເວ 1.6503
0 0 ແກ້ວມະນີ ແກ້ວມະນີ 1.6094

```

0 0 အချိန် ဟိ 1.6094  
 0 0 အချိန် လား 1.6094  
 0 0 ရှိ အချိန် 0.9163  
 0 0 ရှိ ဟိ 0.9163  
 0 0 ရှိ လား 0.9163  
 0 0 လား အချိန် 0.9163  
 0 0 လား ဟိ 0.9163  
 0 0 လား လား 0.9163  
 0 0 ဘာ ပြာ 1.2528  
 0 0 ဘာ လေး 2.0260  
 0 0 ပြာ အ 3.7297  
 0 0 ပြာ ပြာ 0.7419  
 0 0 ပြာ လေး 3.1246  
 0 0 သလဲ အ 2.3434  
 0 0 သလဲ ပြာ 1.4351  
 0 0 သလဲ လေး 2.0260  
 0 0 ဘာ လုပ် 1.2528  
 0 0 လုပ် အ 3.7297  
 0 0 လုပ် လုပ် 0.7419  
 0 0 လုပ် လေး 3.1246  
 0 0 သလဲ လုပ် 1.4351  
 0 0 ဘယ် အ 2.8824  
 0 0 ဘယ် ချိန် 1.1896  
 0 0 ဘယ် လေး 2.9014  
 0 0 အချိန် အ 3.4420  
 0 0 အချိန် ချိန် 0.9383  
 0 0 အချိန် လေး 3.1246  
 0 0 လဲ အ 3.2189  
 0 0 လဲ ချိန် 1.1896  
 0 0 လဲ လေး 2.0260  
 0 0 သူ ယင်းသူ 1.1787  
 0 0 သူ အေမာ 1.1787  
 0 0 သူ နီ 3.0123  
 0 0 သူ လေး 4.5109  
 0 0 ဘယ်မှာ ယင်းသူ 1.1787  
 0 0 ဘယ်မှာ အေမာ 1.1787  
 0 0 ဘယ်မှာ နီ 3.0123  
 0 0 ဘယ်မှာ လေး 4.5109  
 0 0 နေ ယင်းသူ 1.4663  
 0 0 နေ အေမာ 1.4663  
 0 0 နေ နီ 1.4028  
 0 0 နေ လေး 2.7191

0 0 လဲ ယင်းသူ 1.8718  
0 0 လဲ အေမာ 1.8718  
0 0 လဲ နီ 3.0123  
0 0 ဘယ် အိမ် 2.1145  
0 0 ဘယ် မှာ 2.1145  
0 0 ဘယ် မင်း 2.7726  
0 0 ဘယ် နီ 2.5014  
0 0 အိမ် ခာ 3.4420  
0 0 အိမ် အိမ် 1.4955  
0 0 အိမ် မှာ 1.4955  
0 0 အိမ် မင်း 2.1665  
0 0 အိမ် နီ 1.9136  
0 0 အိမ် လေး 3.4122  
0 0 မှာ ခာ 3.4420  
0 0 မှာ အိမ် 1.4955  
0 0 မှာ မှာ 1.4955  
0 0 မှာ မင်း 2.1665  
0 0 မှာ နီ 1.9136  
0 0 မှာ လေး 3.4122  
0 0 မင်း အိမ် 1.6625  
0 0 မင်း မှာ 1.6625  
0 0 မင်း နီ 1.9136  
0 0 မင်း လေး 2.9014  
0 0 နေ့ ခာ 3.4420  
0 0 နေ့ အိမ် 1.8632  
0 0 နေ့ မှာ 1.8632  
0 0 နေ့ မင်း 2.3671  
0 0 သလဲ အိမ် 2.4510  
0 0 သလဲ မှာ 2.4510  
0 0 သလဲ မင်း 2.6184  
0 0 သလဲ နီ 2.5014  
0 0 မင်း တိ 2.7081  
0 0 မင်း စီမံ 2.7081  
0 0 မင်း ဖို့လေး 2.8332  
0 0 ဘာ တိ 1.7918  
0 0 ဘာ စီမံ 1.7918  
0 0 ဘာ ဖို့လေး 1.7346  
0 0 တွေ မင်း 3.8712  
0 0 တွေ ခာ 3.0366  
0 0 တွေ တိ 1.2040  
0 0 တွေ စီမံ 1.2040  
0 0 တွေ ဖို့လေး 1.9169

0 0 စီမံ မင်း 3.8712  
0 0 စီမံ ၁၁ 3.0366  
0 0 စီမံ တိ 1.2040  
0 0 စီမံ စီမံ 1.2040  
0 0 စီမံ ဖို့လေး 1.9169  
0 0 မှာလဲ မင်း 3.8712  
0 0 မှာလဲ ၁၁ 3.0366  
0 0 မှာလဲ တိ 1.7918  
0 0 မှာလဲ စီမံ 1.7918  
0 0 မှာလဲ ဖို့လေး 1.1285  
0 0 မင်း ချက် 2.5257  
0 0 ၁၁ ချက် 1.4271  
0 0 ချက် မင်း 3.8712  
0 0 ချက် ၁၁ 3.4420  
0 0 ချက် ချက် 0.9163  
0 0 ချက် လေး 2.9014  
0 0 လဲ မင်း 4.5643  
0 0 လဲ ချက် 1.2730  
0 0 မင်း မိန္ဒါး 2.3514  
0 0 ၁၁ မိန္ဒါး 1.4351  
0 0 မေး မင်း 3.8712  
0 0 မေး ၁၁ 3.0366  
0 0 မေး မိန္ဒါး 0.8473  
0 0 မေး ဖို့လေး 1.9169  
0 0 မှာလဲ မိန္ဒါး 1.4351  
0 0 မင်း အကြံပီး 2.7300  
0 0 မင်း ချင် 2.7300  
0 0 ၁၁ အကြံပီး 1.6314  
0 0 ၁၁ ချင် 1.6314  
0 0 အကြံပေး မင်း 3.4657  
0 0 အကြံပေး ၁၁ 3.0366  
0 0 အကြံပေး အကြံပီး 1.2637  
0 0 အကြံပေး ချင် 1.2637  
0 0 အကြံပေး လေး 2.5649  
0 0 ချင် မင်း 3.4657  
0 0 ချင် ၁၁ 3.0366  
0 0 ချင် အကြံပီး 1.2637  
0 0 ချင် ချင် 1.2637  
0 0 ချင် လေး 2.5649  
0 0 သလဲ အကြံပီး 1.7492  
0 0 သလဲ ချင် 1.7492

```
0 0 </s> </s> 0  
0
```

Input symbol file ഓന്തോൺ മാലയ്‌ക്ക്

```
[93]: !python extract_column_symbols.py --column 2 < onetoone.txt > onetoone.isym
```

Output symbol file ഓന്തോൺ മാലയ്‌ക്ക്

```
[94]: !python extract_column_symbols.py --column 3 < onetoone.txt > onetoone.osym
```

```
[95]: !fstcompile --isymbols onetoone.isym --osymbols onetoone.osym --keepisymbols  
      --keeposymbols onetoone.txt > onetoone.all.fst
```

```
[96]: !fstdraw --portrait --isymbols=onetoone.isym --osymbols=onetoone.osym ./  
      onetoone.fst | dot -Tpdf -Gmargin=0 > ./onetoone.pdf
```

```
[97]: !pdfcrop onetoone.pdf onetoone_cropped.pdf
```

```
PDFCROP 1.42, 2023/04/15 - Copyright (c) 2002-2023 by Heiko Oberdiek, Oberdiek  
Package Support Group.  
==> 1 page written on `onetoone_cropped.pdf`.
```

```
[100]: !convert -density 1000 onetoone_cropped.pdf -quality 100 onetoone_cropped.png
```

```
[101]: from IPython.display import Image  
#Image(filename="onetoone_cropped.png", height=100, width=300)  
Image(filename="onetoone_cropped.png")
```

```
[101]:
```



```
[102]: !cat onetoone.all.isym
```

```
<eps> 0
မင်္ဂလာ: 1
ဘား 2
အချိန် 3
ရှိ 4
လား 5
ပြော 6
သလဲ 7
လုပ် 8
ဘယ် 9
လဲ 10
သူ 11
ဘယ်မှာ 12
နေ 13
အိမ် 14
မှာ 15
တွေ 16
စီမံ 17
မှာလဲ 18
ချက် 19
ဓာတ် 20
အကြော်း 21
ချင် 22
</s> 23
```

```
[103]: !fstcompose onetoone.fst bigram.fst composed.fst
!echo "fstcompose together a translation model and language model finished!"
```

fstcompose together a translation model and language model finished!

```
[104]: !fstdraw --portrait ./composed.fst | dot -Tpdf -Gmargin=0 > ./composed.pdf
```

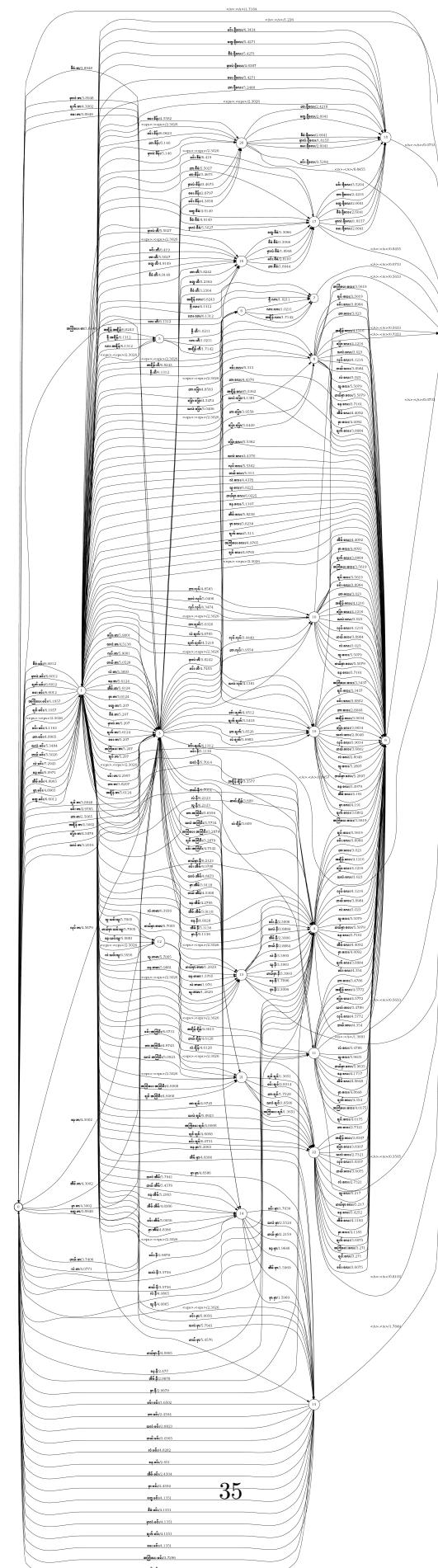
```
[105]: !pdfcrop composed.pdf composed_cropped.pdf
```

PDFCROP 1.42, 2023/04/15 - Copyright (c) 2002-2023 by Heiko Oberdiek, Oberdiek  
Package Support Group.  
==> 1 page written on `composed\_cropped.pdf`.

```
[106]: #!/convert pos_lm_cropped.pdf pos_lm_cropped.png
!convert -density 1000 composed_cropped.pdf -quality 100 composed_cropped.png
```

```
[107]: Image(filename="composed_cropped.png")
```

[107] :



## 1.20 search.fst, searchrmeps.fst, shortest-path.fst

Input օնձատէ թաշկանց տժեցանց զւնդիքայութէ առաջնային shortest-path գիր ըրպիտաց||

```
[110]: !fstdraw --portrait ./search.fst | dot -Tpdf -Gmargin=0 > ./search.pdf
```

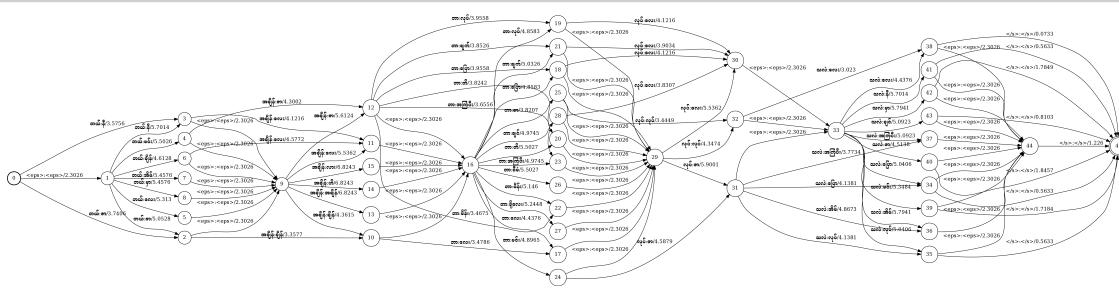
```
[111]: !pdfcrop search.pdf search_cropped.pdf
```

PDFCROP 1.42, 2023/04/15 - Copyright (c) 2002-2023 by Heiko Oberdiek, Oberdiek  
Package Support Group.  
==> 1 page written on `search\_cropped.pdf`.

```
[112]: #!convert pos_lm_cropped.pdf pos_lm_cropped.png  
!convert -density 1000 search_cropped.pdf -quality 100 search_cropped.png
```

```
[116]: Image(filename="search_cropped.png")
```

```
[116]:
```



Ալտհա: տէ ցանց գրաֆ է լինի: Ըստ լաշեցանց||

```
[117]: !fstdraw --portrait ./searchrmeps.fst | dot -Tpdf -Gmargin=0 > ./searchrmeps.pdf
```

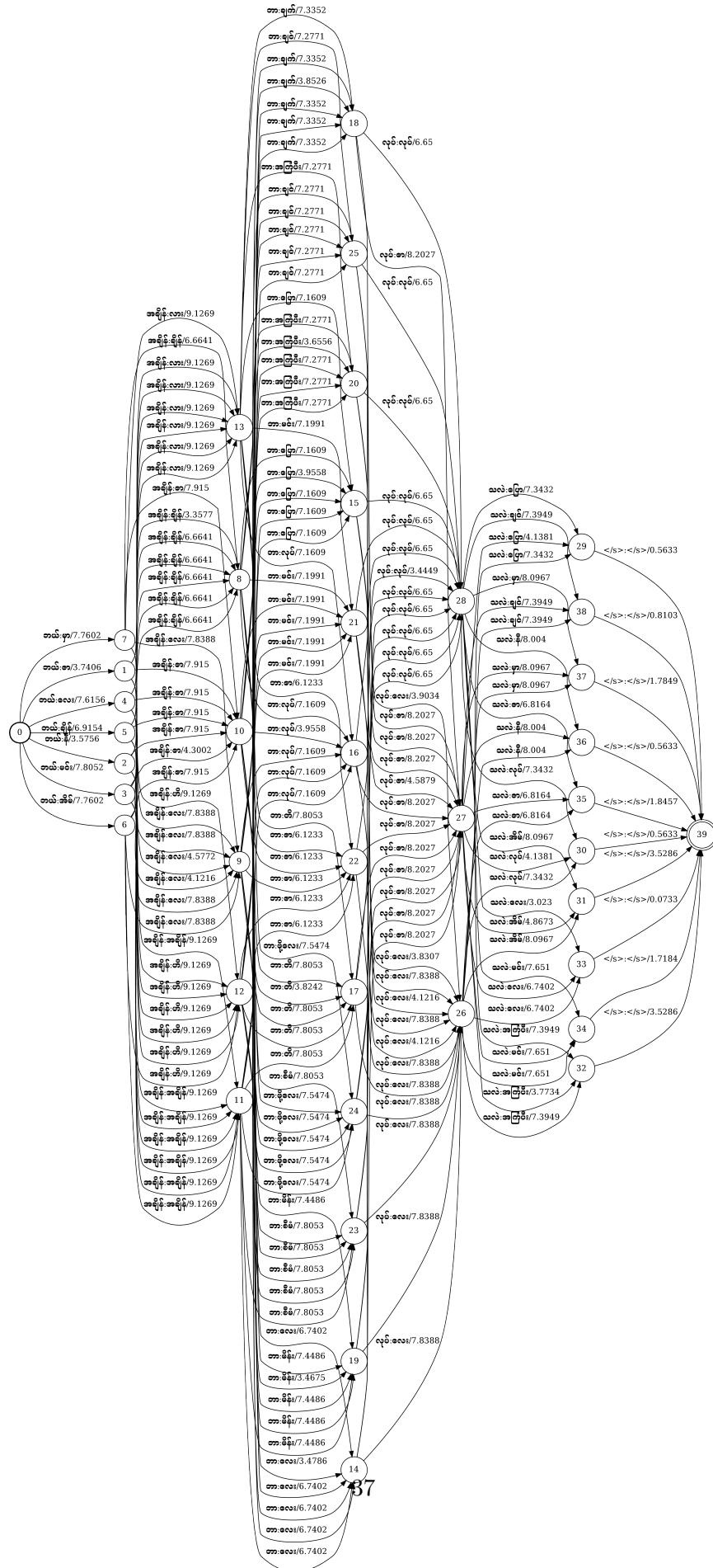
```
[118]: !pdfcrop searchrmeps.pdf searchrmeps_cropped.pdf
```

PDFCROP 1.42, 2023/04/15 - Copyright (c) 2002-2023 by Heiko Oberdiek, Oberdiek  
Package Support Group.  
==> 1 page written on `searchrmeps\_cropped.pdf`.

```
[119]: !convert -density 1000 searchrmeps_cropped.pdf -quality 100 searchrmeps_cropped.png
```

```
[120]: Image(filename="searchrmeps_cropped.png")
```

```
[120]:
```



နောက်ဆုံး shortest-path ကို လေ့လာကြည့်ရအောင်...

```
[121]: !fstdraw --portrait ./shortest-path.fst | dot -Tpdf -Gmargin=0 > ./  
       ↵shortest-path.pdf
```

```
[122]: !pdffcrop shortest-path.pdf shortest-path_cropped.pdf
```

PDFCROP 1.42, 2023/04/15 - Copyright (c) 2002-2023 by Heiko Oberdiek, Oberdiek  
Package Support Group.

==> 1 page written on `shortest-path\_cropped.pdf'.

```
[125]: !convert -density 300 shortest-path_cropped.pdf -quality 100 ↵  
       ↵shortest-path_cropped.png
```

```
[126]: Image(filename="shortest-path_cropped.png")
```

```
[126]:
```

## 1.21 Summary

- WFST framework ဖြစ်တဲ့ OpenFST command တွေနဲ့ IBM Model-1 alignment ရယ်၊ Bigram language model ရယ်ကို သုံးပြီး မြန်မာ-ရှုံးမြန်အကြား machine translation ကို လက်တွေ့လုပ်ပြခဲ့ပါတယ်။
- FST မော်ဒယ်တွေကို graph အနေနဲ့ လေ့လာလို့ရအောင် visualization လုပ်ပြခဲ့ပါတယ်။

## 1.22 References

- Finite State Machines for NLP, Ye Kyaw Thu, 6 Dec 2019, <https://github.com/ye-kyaw-thu/NLP-Class/blob/master/slide/11-fsm4nlp.pdf>
- Machine Translation and Sequence to Sequence Models: [Machine Translation and Sequence to Sequence Models](#)
- OpenFST Library: <https://www.openfst.org/twiki/bin/view/FST/WebHome>
- Anymalign: <https://anymalign.limsi.fr/>
- Juan LUO, Jing SUN, Yves LEPAGE, Improving Sampling-based Alignment Method for Statistical Machine Translation Tasks, IPSJ 17th Annual NLP Conference, 2011, pp. 186-189. [\[Link\]](#)
- IBM Alignment Models: [https://en.wikipedia.org/wiki/IBM\\_alignment\\_models](https://en.wikipedia.org/wiki/IBM_alignment_models)
- BLEU (bilingual evaluation understudy) score: <https://en.wikipedia.org/wiki/BLEU>
- chrF, a tool for calculating character n-gram F score: <https://github.com/m-popovic/chrF>

## 1.23 Citation

If you use this Jupyter Notebook for your teaching or as a baseline for research and development (R&D), please cite the following paper:

Thazin Myint Oo, Thitipong Tanprasert, Ye Kyaw Thu, Thepchai Supnithi, “Transfer and Triangulation Pivot Translation Approaches for Burmese Dialects,” in IEEE Access, vol. 11, pp. 6150-6168, 2023, doi: 10.1109/ACCESS.2023.3236804. (Received 15 October 2022, accepted 27 December 2022, date of publication 13 January 2023, date of current version 20 January 2023.) [\[Link\]](#)

[ ]: