oppaWord

August 3, 2025

0.1 oppaWord

oppaWord demo running or 1st formal experiment
by Ye Kyaw Thu, LU Lab., Myanmar
26 July 2025

[1]:
```
!pwd
```

```
/home/ye/exp/myTokenizer
```

[2]:
```
cd oppaWord
```

```
/home/ye/exp/myTokenizer/oppaWord
```

လောလောဆယ် data ဆိုတဲ့ ဖိုလ်ဒါအောက်ထဲမှာပဲ dictionary, syllable frequency, language model ဖိုင်နဲ့ original corpus နဲ့ တခြား preprocessing လုပ်တုန်းက ဖိုင်တချို့ပါ သိမ်းထားတယ်။

[4]:
```
!tree ./data
```

```
./data
  10k_test.input
  10k_test.txt
  10lines.ref
  myg2p_mypos.dict
  myMono_clean_syl.6gram.arpa
  myMono_clean_syl.6gram.probing.bin
  myMono_clean_syl.6gram.trie.bin
  myMono_clean_syl.arpa
  myMono_clean_syl.probing.bin
  myMono_clean_syl.trie.bin
  myMono.freq
  mypos-ver.3.0.shuf.notag.nopunc.txt.seg_normalized2
  otest.1k.word
  otest.1k.word.input
  prepare_raw
      note.txt
      raw2.txt
      raw3.txt
      raw4.txt
      raw.txt
```

```
    smart_space_remover.py
  raw.txt
  rules.txt

2 directories, 22 files
```

oppaWord ရဲ့ feature အပြည့်နဲ့ word segmentation experiment လုပ်ချင်ရင် တကယ်တမ်း လိုအပ်တာက အောက်ပါ လေးဖိုင်ပါ။
1. myg2p_mypos.dict (dictionary file) 2. myMono.freq (syllable frequency file) 3. Language model file 4. rules.txt (wrong|||correct ဆိုတဲ့ ပုံစံနဲ့ ပြင်ဖို့လိုအပ်တဲ့ utf-8 plain text ဖိုင်ပါ)

0.2 Language Model (LM) File

လောလောဆယ် oppaWord development လုပ်ရင်းနဲ့ 5-gram, 6-gram syllable LM တွေကို ပြင်ပြီး စမ်းကြည့်ခဲ့တယ်။

1. myMono_clean_syl.arpa (5-gram syllable LM, ARPA format)
2. myMono_clean_syl.probing.bin (5-gram syllable LM, probing binary format)
3. myMono_clean_syl.trie.bin (5-gram syllable LM, trie binary format)

0.3 prepare_raw Folder

ဒီ folder အောက်မှာ ရှိတဲ့ ဖိုင်တွေက smart_space_remover.py code ကို ရေးရင်းနဲ့ စမ်းကြည့်ထားတဲ့ ဖိုလ်ဒါပါ။ oppaWord ကို public release လုပ်တဲ့အခါမှာ ပါချင်မှ ပါလိမ့်မယ်။

0.4 Closed-Test Data

1. 10k_test.txt (ရှေ့က baselines.ipynb မှာတုန်းက ပြင်ဆင်ခဲ့တဲ့ closed-test data ပါ)
2. 10k_test.input (oppa_word.py ကို input လုပ်တဲ့အခါမှာ space တွေကို ဖြုတ်ထားရတာမို့ အဲဒီအတွက် ပြင်ဆင်ထားတဲ့ ဖိုင်ပါ)

0.5 Open-Test Data

1. otest.1k.word (myPOS ကနေပဲ ယူလာတဲ့ open-test data ပါ။ POS Tag တွေက ဖြုတ်ထားပြီးသားပါ။)
2. otest.1k.word.input (otest.1k.word ဖိုင်ကိုပဲ oppa_word.py နဲ့ run ဖို့အတွက် space တွေဖြုတ်ထားတဲ့ ဖိုင်ပါ)

ဒီ experiment အတွက် အဓိက လိုအပ်တဲ့ resource file တွေက ဒါအကုန်ပါပဲ။

0.6 File Formats

Binary ဖိုင်မဟုတ်တဲ့ plain text ဖိုင်တွေရဲ့ format တွေကိုလည်း လေ့လာလို့ ရအောင် ရိုက်ထုတ်ပြပါမယ်။

0.6.1 Dictionary File

[5]:
```
!head ./data/myg2p_mypos.dict
```

```
က
ကံ
ကကတစ်
```

ကကတိုး
ကံကျွေး
ကံကျွေးချစနစ်
ကံကြမ္မာ
ကကြိုး
ကကြိုးတန်ဆာ
ကကြီး

[7]:
```
!tail ./data/myg2p_mypos.dict
```

ဩဘာပေး
ဩရသ
ဩဝါဒ
ဩဝါဒခံ
ဩဝါဒါစရိယ
ဩသဓ
ဩအည်းလိုက်
ဪလဲ
။
။

Dictionary ဖိုင်ထဲမှာ ပုဒ်ထီး၊ ပုဒ်မ ကိုလည်း စာလုံးတစ်လုံးအနေနဲ့ သတ်မှတ်ပြီး ထည့်ထားတာကို တွေ့ရပါလိမ့်မယ်။

0.6.2 Syllable Frequency File

ဒီ frequency file က ကျွန်တော်တို့ release လုပ်ဖို့ ပြင်ဆင်နေဆဲဖြစ်တဲ့ Monolingual corpus ကို သုံးပြီး ဆွဲထုတ်ထားတဲ့ frequency file ပါ။ စာကြောင်းရေ တစ်သန်းလေးသောင်းကျော်ကနေ ဆွဲထုတ်ယူထားတာ ဖြစ်ပါတယ်။

[8]:
```
!head ./data/myMono.freq
```

```
2097691 အ
1104050 သည်
796400 ကို
688344 မ
657077 ပါ
613841 က
536481 ဖြစ်
517064 များ
478096 ရ
426865 နေ
```

[10]:
```
!wc ./data/myMono.freq
```

```
 21845  43692 412595 ./data/myMono.freq
```

0.6.3 Language Model File

Language Model ဖိုင်တွေက ARPA format အနေနဲ့ သိမ်းထားတာရယ်၊ binary file အနေနဲ့ သိမ်းထားတာရယ်ဆိုပြီး ရှိလိမ့်မယ်။ Binary file ကိုမှာ probing model နဲ့ trie model ဆိုပြီး နှစ်မျိုးသိမ်းလို့ ရပါတယ်။ Probing model က linear probing hash table ကို သုံးပြီးတော့ probing LM က trie lM ထက် စာရင် 81% ပိုမြန်ပါတယ်။ ဒါပေမဲ့ memory ကိုပိုသုံးလိမ့်မယ်။ Trie နဲ့ သိမ်းထားတဲ့ LM ကတော့ bit level packing လုပ်ထားတဲ့ trie structure ပါ။ အဲဒါကြောင့် သူက probing ထက်စာရင် memory သုံးတာ သက်သာပါလိမ့်မယ်။ သို့သော်လည်း n-gram ဒေတာတွေကို ရှာဖွေတဲ့အခါမှာတော့ probing ထက် နှေးပါလိမ့်မယ်။ အသေးစိတ် သိချင်တဲ့ သူတွေက KenLM paper ကို ဖတ်ကြည့်ပါ။

[11]:
```
!head -n 30 ./data/myMono_clean_syl.arpa
```

```
\data\
ngram 1=21849
ngram 2=890674
ngram 3=6273833
ngram 4=14881991
ngram 5=22249448

\1-grams:
-5.9508724      <unk>   0
0       <s>     -2.3404233
-2.5402012      </s>    0
-2.969469       လေ့      -1.0170939
-3.2468214      ကျင့်    -0.7553899
-3.1590524      ခန်း     -0.76661134
-2.8109293      လုပ်     -1.2248719
-2.7846105      ခြင်း    -1.2950884
-2.273433       အ        -1.630708
-2.7302542      လေး      -1.122314
-2.474841       မ        -1.388715
-2.7878892      ဟာ       -1.0509427
-3.393969       ယောကျ်ား         -0.82077605
-3.1278503      ဝီ       -0.70778507
-2.5439384      သ        -1.138216
-2.798183       စေ       -1.0846236
-2.9547236      ဖို့     -1.0897424
-2.6649404      နဲ့      -1.278924
-2.8685365      ကိုယ်    -1.0504222
-3.3625221      ခန္ဓာ    -0.7001066
-3.622274       တောင့်   -0.6469929
-3.238298       တင်း     -0.61249375
```

အထက်ပါ မြင်ရတဲ့အတိုင်း ARPA format မှာက header ပိုင်းပါလိမ့်မယ်။ အဲဒီကနေ ဘယ်လောက် gram LM လဲ ဆိုတာကို သိရလိမ့်မယ်။ ပြီးတော့ Unigram ကနေ ဆောက်ထားတဲ့ ngram အထိ syllable အတွဲတွေရဲ့ information ကို လေ့လာနိုင်တယ်။

syllable တစ်လုံး သို့မဟုတ် syllable အတွဲတွေရဲ့ information ကိုတော့ အောက်ပါ format နဲ့ သိမ်းထားပါတယ်။

```
log10_probability    word    backoff_weight
```

0.6.4 Rules File

Rule ဖိုင်ကတော့ oppa_word ကနေ segmentation ဖြတ်ပြီးထွက်လာတဲ့ အမှားတွေကို ကြည့်ပြီး rule ထုတ်ပြီးတော့ ပြင်ချင်တဲ့အခါမှာ သုံးတဲ့ ဖိုင်ပါ။ ဥပမာအနေနဲ့ ဆောက်ပြီးတော့ သုံးပြထားတာပါ။ ကိုယ် word segmentation လုပ်ချင်တဲ့ ဒိုမိန်း အကုန်ကို oppaWord က မှန်မှန်ကန်ကန် ဖြတ်ပေးနိုင်မှာ မဟုတ်ပါဘူး။ ဥပမာ ဆေးပညာဒိုမိန်း၊ ရူပဗေဒဒိုမိန်း မှာ ပါတဲ့ စာလုံးမျိုးတွေပါ။ အဲဒီလို ဒိုမိန်းအတွက် စာလုံးဖြတ်တာကို မှန်ကန်ဖို့အတွက် ဆိုရင် oppaWord ကို တစ်ခေါက် run ကြည့်လိုက်ပြီး မှားဖြတ်ထားတဲ့ စာလုံးတွေကို ပြင်ချင်ရင် rule ဖိုင်ပြင်ပြီး အလွယ်တကူ ပြင်ဆင်လို့ ရပါတယ်။

[12]:
```
!cat ./data/rules.txt
```

```
ပါတယ်|||ပါ တယ်
မရှိ|||မ ရှိ
ဒီနေ့|||ဒီ နေ့
ဖြစ်သည်|||ဖြစ် သည်
ခဲ့သည်|||ခဲ့ သည်
သူက|||သူ က
မဟုတ်|||မ ဟုတ်
များကို|||များ ကို
ပါဘူး|||ပါ ဘူး
ကတော့|||က တော့
ရှိတယ်|||ရှိ တယ်
ချင်ပါ|||ချင် ပါ
ချင်တယ်|||ချင် တယ်
မြန်မာနိုင်ငံ|||မြန်မာ နိုင်ငံ
တို့၏|||တို့ ၏
ပါသလဲ|||ပါ သလဲ
မှာရှိ|||မှာ ရှိ
များ၏|||များ ၏
ထင်တယ်|||ထင် တယ်
မှာထား|||မှာ ထား
စီးပွားရေး|||စီးပွား ရေး
အလုပ်လုပ်|||အလုပ် လုပ်
တာနဲ့|||တာ နဲ့
လိုက်ပါ|||လိုက် ပါ
ဒီဟာ|||ဒီ ဟာ
သူများ|||သူ များ
သူ၏|||သူ ၏
ဟုတ်လား|||ဟုတ် လား
တစ်ဦး|||တစ် ဦး
```

နံ ပါတ်|||နံပါတ်
(\S)([။]){3}\1 \2

Rule file format က wrong|||correct ဆိုတဲ့ format ပါ။
Rule ဖိုင်ထဲမှာ အထက်မှာ ဥပမာအနေနဲ့ ပြထားတဲ့အတိုင်း Regular Expression (RE) တွေကိုလည်း ထည့်သုံးလို့ ရအောင် oppaWord က support ပေးထားပါတယ်။ :)

(\S)([။]){3}\1 \2 RE ကတော့ ပုဒ်ထီး၊ ပုဒ်မ သင်္ကေတတွေနဲ့ စာလုံးက ပူးကပ်နေရင် space ခြားဖို့ ရေးထားတာပါ။

0.6.5 Closed, Open Test Data Format

Experiment တစ်ခု လုပ်ပြီဆိုရင် ထုံးစံအတိုင်း closed, open test data တွေကို ပြင်ဆင်ရပါတယ်။ Closed ကတော့ မော်ဒယ် ကို training လုပ်စဉ်က သုံးထားတဲ့ ဒေတာထဲကနေပဲ ဆွဲထုတ်ယူထားတဲ့ ဒေတာကို ပြောတာပါ။ Open test data ကတော့ မော်ဒယ် training လုပ်စဉ်မှာ မပါတဲ့ ဒေတာ၊ တနည်းအားဖြင့် မော်ဒယ်က မမြင်ဘူးသေးတဲ့ ဒေတာ အသစ်ပါ။ ကိုယ်ဆောက်ထားတဲ့ မော်ဒယ်က ဘယ်လောက်ထိ အလုပ်လုပ်ပေးနိုင်သလဲ ဆိုသာကို တိုင်းတာဖို့ အတွက်က closed ရော၊ open ရော နှစ်မျိုးစလုံးက အရေးကြီးပါတယ်။ အဲဒီနှစ်မျိုးထဲကမှ ပိုအရေးကြီးတာက ဘယ်ဟာလဲ ဆိုရင်တော့ open test data လို့ ဖြေရမှာပေါ့။

oppa_word က တကယ်တမ်းက LM ကလွဲရင် သပ်သပ်ကြီး training လုပ်ယူစရာ မလိုပါဘူး။ တကယ်ကို အဘိဓာန်ကို သုံး၊ syllable frequency နဲ့ LM ကို သုံးပြီး scoreing လုပ်ပြီး မြန်မြန်ဆန်ဆန် word segmentation လုပ်ပေးနိုင်အောင် ရေးထားတာပါ။ အဲဒါကြောင့် training လုပ်စဉ်က သုံးခဲ့တဲ့ ဒေတာလို့ တိုက်ရိုက်တော့ သုံးလို့ မရပါဘူး။ သို့သော်လည်း အဘိဓာန်ထဲမှာ myG2P အပြင် myPOS က စာလုံးတွေကိုပါ ဖြည့်ထားတာမို့ myPOS data တွေက ပါနေပါတယ်။ အဲဒါကြောင့် open test data ကို ပြင်တဲ့အခါမှာ myPOS ရဲ့ open test data ကိုပဲ ယူသုံးကြည့်ထားတာပါ။

သိကြတဲ့အတိုင်း မြန်မာစာအတွက်က manual word segmentation လုပ်ထားတဲ့ ဒေတာကလည်း ရှားတာမို့လို့...

[14]:
```
!head ./data/otest.1k.word
```

တစ် ကိုက် ကို ဝမ် ခုနှစ်ထောင် ပါ ။
မနှစ် က သူ ကျွန်မ ကို သင် ပေး တယ် ။
ကျွန်တော့် ခုံ သွား ရှာ မလို့ ။
အတန်း စ တာ ကြာ ပြီ လား ။
ဆေး နည်းနည်း စား လိုက် ၊ သုံး လေး ရက် လောက် အနားယူ လိုက် ရင် ပျောက် သွား မှာ ပါ ။
အေးချမ်း မှု နဲ့ စည်းကမ်း ကို တည်မြဲ အောင် ထိန်းသိမ်း သည် ။
ဇွန်း ကို လိုအပ် တယ် ။
ဘွဲ့ ရ ရင် ဘာ လုပ် မ လို့ လဲ ။
ကျွန်တော် ချောင်းဆိုး ခြင်း အတွက် တစ် ခု ခု လို ချင် တယ် ။
အသီးအနှံ တို့ မှ လွဲ လျှင် လူ တို့ ၏ အဓိက အစားအစာ မှာ ငါး ဖြစ် သည် ။

[15]:
```
!head ./data/otest.1k.word.input
```

တစ်ကိုက်ကိုဝမ်ခုနှစ်ထောင်ပါ။
မနှစ်ကသူကျွန်မကိုသင်ပေးတယ်။

ကျွန်တော့်ခုံသွားရှာမလို့။
အတန်းစတာကြာပြီလား။
ဆေးနည်းနည်းစားလိုက်၊သုံးလေးရက်လောက်အနားယူလိုက်ရင်ပျောက်သွားမှာပါ။
အေးချမ်းမှုနဲ့စည်းကမ်းကိုတည်မြဲအောင်ထိန်းသိမ်းသည်။
ဇွန်းကိုလိုအပ်တယ်။
ဘွဲ့ရရင်ဘာလုပ်မလို့လဲ။
ကျွန်တော်ချောင်းဆိုးခြင်းအတွက်တစ်ခုခုလိုချင်တယ်။
အသီးအနှံတို့မှလွဲလျှင်လူတို့၏အဓိကအစားအစာမှာငါးဖြစ်သည်။

[16]:
```
!tail ./data/otest.1k.word
```

အိုး ခွက် ပန်း ကန် တွေ သိပ် မ ရှိ လို့ ထမင်း ဟင်း ချက် ရ တာ အဆင်မပြေ ဘူး ။
စိတ် ဝင်စား ဖို့ ကောင်း တယ် ။
ဒီ ဆေး ကို တဝက် စီ ခွဲ ပေး ပါ ။
ရောင်း ကောင်း လား ။
ဆရာ ဒီ သွား က ခဏခဏ နာ နေ တယ် ။
အခု ဘာ လုပ် နေ လဲ ။
ဇူလိုင် ၁၄ ရက် မှာ ဘန်ကောက် ကို သွား မယ့် US 123 မှာ ပါ ။ ဟုတ် လား ။
ကား မှ ကားဘီး ကို ဖြုတ် လိုက် သည် ။
ကျွန်တော် သိ ပါရစေ ။
ဘူတာရုံ က အလွန်တရာ ပြည့်ကျပ် နေ သည် ။

[17]:
```
!tail ./data/otest.1k.word.input
```

အိုးခွက်ပန်းကန်တွေသိပ်မရှိလို့ထမင်းဟင်းချက်ရတာအဆင်မပြေဘူး။
စိတ်ဝင်စားဖို့ကောင်းတယ်။
ဒီဆေးကိုတဝက်စီခွဲပေးပါ။
ရောင်းကောင်းလား။
ဆရာဒီသွားကခဏခဏနာနေတယ်။
အခုဘာလုပ်နေလဲ။
ဇူလိုင် ၁၄ ရက်မှာဘန်ကောက်ကိုသွားမယ့် US 123 မှာပါ။ဟုတ်လား။
ကားမှကားဘီးကိုဖြုတ်လိုက်သည်။
ကျွန်တော်သိပါရစေ။
ဘူတာရုံကအလွန်တရာပြည့်ကျပ်နေသည်။

oppa_word.py နဲ့ run ဖို့အတွက် input ဖိုင်ကိုတော့ smart_space_remover.py ကို သုံးထားပါတယ်။

0.7 smart_space_remover.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""
smart_space_remover.py: Remove spaces intelligently for Myanmar text segmentation.
written by Ye Kyaw Thu, LU Lab., Myanmar.
```

```
last update: 25 July 2025

Modes:
  - all      : Remove all spaces
  - my       : Remove spaces only between Myanmar letters
  - my_not_num  : Like 'my' but preserve spacing near Myanmar numbers

Usage:
  $ python smart_space_remover.py --mode my_not_num --input input.txt --output output.txt
"""

import sys
import argparse
import re

# === Unicode Character Classes ===
MYANMAR_LETTER = r'[\u1000-\u109F\uAA60-\uAA7F]'
MYANMAR_DIGIT = r'[\u1040-\u1049]'

# === Regex Patterns ===
RE_MM_LETTER_SPACE = re.compile(rf'({MYANMAR_LETTER})\s+({MYANMAR_LETTER})')

# Match MyanmarDigit <space> MyanmarDigit
RE_MM_DIGIT_DIGIT = re.compile(rf'({MYANMAR_DIGIT})\s+({MYANMAR_DIGIT})')

# Match MyanmarDigit <space> MyanmarLetter
RE_MM_DIGIT_LETTER = re.compile(rf'({MYANMAR_DIGIT})\s+({MYANMAR_LETTER})')

# Match MyanmarLetter <space> MyanmarDigit
RE_MM_LETTER_DIGIT = re.compile(rf'({MYANMAR_LETTER})\s+({MYANMAR_DIGIT})')

# Protect standalone Myanmar digit tokens and spacing
PROTECT_SPACES = [
    (RE_MM_DIGIT_DIGIT, r'\1 \2'),           # protect digit-digit
    (RE_MM_DIGIT_LETTER, r'\1 \2'),          # protect digit-letter
    (RE_MM_LETTER_DIGIT, r'\1 \2'),          # protect letter-digit
]

def remove_all_spaces(text):
    return text.replace(' ', '')

def remove_myanmar_spaces(text, preserve_digits=False):
    if preserve_digits:
        # Step 1: Protect spaces between digits and letters
        for pattern, replacement in PROTECT_SPACES:
            text = pattern.sub(replacement, text)

    # Step 2: Remove space between Myanmar letters
```

```
    prev = None
    while prev != text:
        prev = text
        text = RE_MM_LETTER_SPACE.sub(r'\1\2', text)

    if preserve_digits:
        # Step 3: Restore protected spaces
        text = text.replace(' ', ' ')

    return text

def process_lines(lines, mode):
    for line in lines:
        line = line.rstrip('\n')
        if mode == 'all':
            yield remove_all_spaces(line)
        elif mode == 'my':
            yield remove_myanmar_spaces(line, preserve_digits=False)
        elif mode == 'my_not_num':
            yield remove_myanmar_spaces(line, preserve_digits=True)
        else:
            raise ValueError(f"Unknown mode: {mode}")

def main():
    parser = argparse.ArgumentParser(description="Smart Myanmar space remover")
    parser.add_argument('--mode', choices=['all', 'my', 'my_not_num'], required=True,
                        help="Mode: 'all', 'my', or 'my+num'")
    parser.add_argument('--input', type=str, help="Input file (default: stdin)")
    parser.add_argument('--output', type=str, help="Output file (default: stdout)")
    args = parser.parse_args()

    input_stream = open(args.input, 'r', encoding='utf-8') if args.input else sys.stdin
    output_stream = open(args.output, 'w', encoding='utf-8') if args.output else sys.stdout

    try:
        for processed in process_lines(input_stream, args.mode):
            output_stream.write(processed + '\n')
    finally:
        if args.input:
            input_stream.close()
        if args.output:
            output_stream.close()

if __name__ == '__main__':
    main()
```

0.8 –help of oppaWord

oppaWord ကို မြန်မာစာလုံးတွေကို မှန်မှန်ကန်ကန် ဖြတ်ပေးနိုင်ဖို့ ဘယ်လို ဒီဇိုင်းလုပ်ထားခဲ့တယ် ဆိုတာကို သပ်သပ် slide နဲ့ပဲ ရှင်းပါမယ်။ ဒီ Notebook မှာတော့ လက်တွေ့ သုံးပုံသုံးနည်းကိုလည်း ပြရင်းနဲ့ closed-test, open-test ဒေတာတွေကို သုံးပြီး formal evaluation လုပ်တာကိုပဲ အာရုံစိုက်ကြရအောင်။

oppa_word.py code မှာ support လုပ်ပေးထားတဲ့ command line argument တွေကို ကြည့်ချင်ရင် –help နဲ့ ခေါ်ကြည့်ပါ။

[21]:
```
!python oppa_word.py --help
```

```
usage: oppa_word.py [-h] --input INPUT [--output OUTPUT] --dict DICT
                    [--sylfreq SYLFREQ] [--arpa ARPA]
                    [--postrule-file POSTRULE_FILE] [--max-order MAX_ORDER]
                    [--dict-weight DICT_WEIGHT] [--use-bimm-fallback]
                    [--bimm-boost BIMM_BOOST] [--visualize-dag]
                    [--dag-output-dir DAG_OUTPUT_DIR]
                    [--space-remove-mode {all,my,my_not_num}]
                    [--max-word-len MAX_WORD_LEN]

oppa_word, Hybrid DAG + BiMM + LM Myanmar Word Segmenter with optional Aho-
Corasick support

options:
  -h, --help            show this help message and exit
  --input INPUT, -i INPUT
                        Input file with one sentence per line (UTF-8)
  --output OUTPUT, -o OUTPUT
                        Optional output file path (default: stdout)
  --dict DICT, -d DICT  Word dictionary file (one word per line)
  --sylfreq SYLFREQ, -s SYLFREQ
                        Syllable frequency file (syllable<TAB>frequency, for
                        scoring)
  --arpa ARPA, -a ARPA  ARPA-format syllable-level language model (optional)
  --postrule-file POSTRULE_FILE
                        Optional post-processing rules (e.g., merging,
                        corrections)
  --max-order MAX_ORDER
                        Max LM n-gram order (default: 5)
  --dict-weight DICT_WEIGHT
                        Dictionary path weight in scoring (default: 10.0)
  --use-bimm-fallback   Enable Bi-directional Maximum Matching as fallback
  --bimm-boost BIMM_BOOST
                        Boost score added to Bi-MM fallback path (default:
                        0.0)
  --visualize-dag       Generate DAG visualization (PDF per sentence)
  --dag-output-dir DAG_OUTPUT_DIR
                        Directory to save DAG PDFs if --visualize-dag is used
```

```
                        (default: 'dag_viz')
  --space-remove-mode {all,my,my_not_num}
                        Preprocessing mode to remove spaces: 'all', 'my'
                        (Myanmar only), or 'my_not_num (Myanmar but not
                        including Myanmar numbers'
  --max-word-len MAX_WORD_LEN
                        Maximum word length in syllables (3-12, default:6)
```

0.9 1.dict_sylfreq (closed-test)

Dictionary ရယ် syllable frequency ရယ် နှစ်မျိုးကို သုံးပြီး word segmentation လုပ်မယ်။

[26]:
```
!mkdir exp_1
```

[27]:
```
!time python oppa_word.py \
  --input "./data/10k_test.txt" \
  --output "exp_1/dict_sylfreq.seg" \
  --postrule-file "./data/rules.txt" \
  --space-remove-mode "my_not_num" \
  --dict "./data/myg2p_mypos.dict" \
  --sylfreq "./data/myMono.freq"
```

```
real    0m0.758s
user    0m0.745s
sys     0m0.013s
```

အထက်မှာ မြင်ရတဲ့အတိုင်း စာကြောင်းရေ တစ်သောင်းကို word segmentation ဖြတ်တာ real က 0.758 seconds မို့လို့ ၁ စက္ကန့်တောင် မကြာလိုက်ပါဘူး။ Super fast word segmenter ပါ။

[28]:
```
!head ./exp_1/dict_sylfreq.seg
```

```
၁၉၆၂ ခု နှစ် ခန့် မှန်း သန်း ခေါင် စာ ရင်း အ ရ လူဦး ရေ ၁၁၅၉၃၁ ယောက် ရှိ သည်
လူ တိုင်း တွင် သင့် မြတ် လျော် ကန် စွာ ကန့် သတ် ထား သည့် အ လုပ် လုပ် ချိန် အ
ပြင် လ စာ နှင့် တ ကွ အ ခါ ကာ လ အား လျော် စွာ သတ် မှတ် ထား သည့် အ လုပ် ␣
↪အား လပ်
ရက် များ ပါ ဝင် သည့် အ နား ယူ ခွင့် နှင့် အား လပ် ခွင့် ခံ စား ပိုင် ခွင့် ရှိ
သည်
ဤ နည်း ကို စစ် ယူ သော နည်း ဟု ခေါ် သည်
စာ ပြန် ပွဲ ဆို တာ က အာ ဂုံ ဆောင် အ လွတ် ကျက် ထား တဲ့ ပိ ဋ္ဌ ကတ် သုံး ပုံ စာ ပေ
တွေ ကို စာ စစ် သံ ဃာ တော် ကြီး တွေ ရဲ့ ရှေ့ မှာ အ လွတ် ပြန် ပြီး ရွတ် ပြ ရ တာ
ပေါ့
ဒီ မှာ ကျွန် တော့် သက် သေ ခံ ကတ် ပါ
၂ ဝ ရာ စု မြန် မာ့ သ မိုင်း သန်း ဝင်း လှိုင် ၂ ဝ ဝ၉ ခု မေ လ ကံ ကော် ဝတ် ရည် စာ
ပေ
ကျွန် တော် မျက် မှန် တစ် လက် လုပ် ချင် ပါ တယ်
ကျွန် တော် တို့ က ဒီ အ မှု ရဲ့ ကြဲ ရာ ပါ ကို ဖမ်း မိ ဖို့ ကြိုး စား ခဲ့ တယ်
```

က လေး မီး ဖွား ဖို့ ခ နှ် မှန်း ရက် က ဘယ် တော့ ပါ လဲ
အ ရိုး ရှင်း ဆုံး ကာ ဗို ဟိုက် ဒ ရိတ် မှာ ဂ လူး ကို့ စ် ဂ လက် တို့ စ် ဖ ရပ် တို့ စ်
စ သည့် မို နို ဆက် က ရိုက် များ ဖြစ် သည်

0.10 Evaluation for dict_sylfreq (closed-test)

ဒီတစ်ခေါက် oppa_word.py ကို develop လုပ်ရင်းနဲ့ word segmentation အတွက် evaluation ကိုလည်း အသေးစိတ် analysis လုပ်ချင်လို့ eval_segmentation.py ကိုပါ ရေးဖြစ်ခဲ့ပါတယ်။ အဲဒီ ပရိဂရမ်ထဲမှာပါတဲ့ evaluation လုပ်တဲ့အပိုင်းကိုလည်း သပ်သပ် slide ပြင်ပြီး ရှင်းပါမယ်။

[29]:
```
!python ./tools/eval_segmentation.py --help
```

```
usage: eval_segmentation.py [-h] -r REFERENCE [-H HYPOTHESIS] [--top-k TOP_K]
                            [--no-errors]

Enhanced Word Segmentation Evaluator with Error Analysis

options:
  -h, --help            show this help message and exit
  -r REFERENCE, --reference REFERENCE
                        Reference (gold standard) file (default: None)
  -H HYPOTHESIS, --hypothesis HYPOTHESIS
                        Hypothesis (system output) file (use - for stdin)
                        (default: -)
  --top-k TOP_K         Show top K most frequent errors (default: 10)
  --no-errors           Skip error analysis to save time (default: False)
```

[31]:
```
!python ./tools/eval_segmentation.py -r ./data/10k_test.txt -H ./exp_1/
↪dict_sylfreq.seg --top-k 10
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                 0.3122
Word Recall                    0.4516
Word F1-score                  0.3692
------------------------------------------------------------
Boundary Precision             0.0805
Boundary Recall                0.1165
Boundary F1-score              0.0952
------------------------------------------------------------
Vocab Precision                0.5420
Vocab Recall                   0.2012
Vocab F1-score                 0.2935
============================================================
```

```
Additional Statistics:
Reference words: 117857
Hypothesis words: 170501
Correct words: 53230
Reference vocabulary size: 10840
Hypothesis vocabulary size: 4024
Common vocabulary: 2181

Top Segmentation Errors Analysis
============================================================
Total errors: 106473

Most Frequent Over-Segmentation Errors (System split where it shouldn't):

Most Frequent Under-Segmentation Errors (System joined what should be separate):

Most Frequent Complex Boundary Errors:
  592 × REF: 'ကျွန်တော်' → HYP: 'ကျွန်'
  579 × REF: 'ခင်ဗျား' → HYP: 'ခင်'
  198 × REF: 'သည်' → HYP: 'အ'
  175 × REF: 'တို့' → HYP: 'တော်'
  140 × REF: 'ကို' → HYP: 'အ'
  134 × REF: 'အဲဒီ' → HYP: 'အဲ'
  124 × REF: 'ကျွန်တော့်' → HYP: 'ကျွန်'
  124 × REF: 'တယ်' → HYP: 'ပါ'
  120 × REF: 'ပါ' → HYP: 'အ'
  107 × REF: 'ဘူး' → HYP: 'မ'
```

0.11 2.dict_sylfreq_bimmfallback (closed-test)

ဒီတခါတော့ bimmfallback ကိုပါ သုံးပါမယ်။

[33]:
```
!time python oppa_word.py \
  --input ./data/10k_test.txt \
  --output "exp_1/dict_sylfreq_bimmfallback.seg" \
  --postrule-file "./data/rules.txt" \
  --space-remove-mode "my_not_num" \
  --dict "./data/myg2p_mypos.dict" \
  --sylfreq "./data/myMono.freq" \
  --use-bimm-fallback
```

```
real    0m1.075s
user    0m1.067s
sys     0m0.008s
```

[34]:
```
!head ./exp_1/dict_sylfreq_bimmfallback.seg
```

၁၉၆၂ ခု နှစ် ခန့် မှန်း သန်း ခေါင် စာ ရင်း အ ရ လူဦး ရေ ၁၁၅၃၃၁ ယောက် ရှိ သည်
လူ တိုင်း တွင် သင့် မြတ် လျော် ကန် စွာ ကန့် သတ် ထား သည့် အ လုပ် လုပ် ချိန် အ
ပြင် လ စာ နှင့် တ ကွ အ ခါ ကာ လ အား လျော် စွာ သတ် မှတ် ထား သည့် အ လုပ်
↪အား လပ်
ရက် များ ပါ ဝင် သည့် အ နား ယူ ခွင့် နှင့် အား လပ် ခွင့် ခံ စား ပိုင် ခွင့် ရှိ
သည်
ဤ နည်း ကို စစ် ယူ သော နည်း ဟု ခေါ် သည်
စာ ပြန် ပွဲ ဆို တာ က အာ ဂုံ ဆောင် အ လွတ် ကျက် ထား တဲ့ ပိ ဋ္ဌ ကတ် သုံး ပုံ စာ ပေ
တွေ ကို စာ စစ် သံ ဃာ တော် ကြီး တွေ ရဲ့ ရှေ့ မှာ အ လွတ် ပြန် ပြီး ရွတ် ပြ ရ တာ
ပေါ့
ဒီ မှာ ကျွန် တော်် သက် သေ ခံ ကတ် ပါ
၂ ဝ ရာ စု မြန် မာ့ သ မိုင်း သန်း ဝင်း လှိုင် ၂ ဝ ဝ၉ ခု မေ လ ကံ ကော် ဝတ် ရည် စာ
ပေ
ကျွန် တော် မျက် မှန် တစ် လက် လုပ် ချင် ပါ တယ်
ကျွန် တော် တို့ က ဒီ အ မှု ရဲ့ ကြံ ရာ ပါ ကို ဖမ်း မိ ဖို့ ကြိုး စား ခဲ့ တယ်
က လေး မီး ဖွား ဖို့ ခ နှ် မှန်း ရက် က ဘယ် တော့ ပါ လဲ
အ ရိုး ရှင်း ဆုံး ကာ ဗို ဟိုက် ဒ ရိတ် မှာ ဂ လူး ကို့စ် ဂ လက် တို့စ် ဖ ရပ် တို့စ်
စ သည့် မို နို ဆက် က ရိုက် များ ဖြစ် သည်

0.12 Evaluation for dict_sylfreq_bimmfallback (closed-test)

[35]:
```
!python ./tools/eval_segmentation.py -r ./data/10k_test.txt -H ./exp_1/
↪dict_sylfreq_bimmfallback.seg --top-k 10
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                 0.3122
Word Recall                    0.4516
Word F1-score                  0.3692
------------------------------------------------------------
Boundary Precision             0.0805
Boundary Recall                0.1165
Boundary F1-score              0.0952
------------------------------------------------------------
Vocab Precision                0.5420
Vocab Recall                   0.2012
Vocab F1-score                 0.2935
============================================================

Additional Statistics:
Reference words: 117857
Hypothesis words: 170501
```

```
Correct words: 53230
Reference vocabulary size: 10840
Hypothesis vocabulary size: 4024
Common vocabulary: 2181

Top Segmentation Errors Analysis
============================================================
Total errors: 106473

Most Frequent Over-Segmentation Errors (System split where it shouldn't):

Most Frequent Under-Segmentation Errors (System joined what should be separate):

Most Frequent Complex Boundary Errors:
  592 × REF: 'ကျွန်တော်' → HYP: 'ကျွန်'
  579 × REF: 'ခင်ဗျား' → HYP: 'ခင်'
  198 × REF: 'သည်' → HYP: 'အ'
  175 × REF: 'တို့' → HYP: 'တော်'
  140 × REF: 'ကို' → HYP: 'အ'
  134 × REF: 'အဲဒီ' → HYP: 'အဲ'
  124 × REF: 'ကျွန်တော့်' → HYP: 'ကျွန်'
  124 × REF: 'တယ်' → HYP: 'ပါ'
  120 × REF: 'ပါ' → HYP: 'အ'
  107 × REF: 'ဘူး' → HYP: 'မ'
```

ရလဒ်က အပြောင်းအလဲ မရှိဘူး။

0.13 3.dict_sylfreq_bimmfallback_bimmboost50 (closed-test)

[38]:
```
!time python oppa_word.py \
  --input ./data/10k_test.txt \
  --output "exp_1/dict_sylfreq_bimmfallback_bimmboost50.seg" \
  --postrule-file "./data/rules.txt" \
  --space-remove-mode "my_not_num" \
  --dict "./data/myg2p_mypos.dict" \
  --sylfreq "./data/myMono.freq" \
  --use-bimm-fallback \
  --bimm-boost 50
```

```
real    0m1.078s
user    0m1.067s
sys     0m0.010s
```

[43]:
```
!head ./exp_1/dict_sylfreq_bimmfallback_bimmboost50.seg
```

၁၉၆၂ ခုနှစ် ခန့်မှန်း သန်း ခေါင် စာ ရင်း အရ လူဦး ရေ ၁၁၅၃၁၁ ယောက် ရှိ သည်

လူ တိုင်း တွင် သင့်မြတ် လျော်ကန် စွာ ကန့်သတ် ထား သည့် အ လုပ် လုပ် ချိန် အပြင်
လစာ နှင့်တကွ အခါ ကာလ အားလျော်စွာ သတ်မှတ် ထား သည့် အလုပ် အားလပ်ရက် များ ပါဝင်
သည့် အ နား ယူ ခွင့် နှင့် အားလပ်ခွင့် ခံ စား ပိုင် ခွင့် ရှိ သည်
ဤ နည်း ကို စစ်ယူ သော နည်း ဟု ခေါ် သည်
စာပြန်ပွဲ ဆို တာ က အာဂုံဆောင် အလွတ်ကျက် ထား တဲ့ ပိ ဋ္ဌ ကတ် သုံး ပုံ စာပေ တွေ ကို
စာစစ် သံဃာတော်ကြီး တွေ ရဲ့ ရှေ့မှာ အလွတ် ပြန် ပြီး ရွတ်ပြ ရ တာ ပေါ့
ဒီ မှာ ကျွန်တော့် သက် သေ ခံ ကတ် ပါ
၂ ဝ ရာစု မြန်မာ့ သမိုင်း သန်း ဝင်း လှိုင် ၂ ဝ ဝ၉ ခု မေ လ ကံ ကော် ဝတ် ရည် စာပေ
ကျွန်တော် မျက်မှန် တစ် လက်လုပ် ချင် ပါ တယ်
ကျွန်တော် တို့ က ဒီ အမှု ရဲ့ ကြံရာပါ ကို ဖမ်းမိ ဖို့ ကြိုးစား ခဲ့ တယ်
ကလေး မီးဖွား ဖို့ ခန့်မှန်း ရက် က ဘယ်တော့ ပါ လဲ
အ ရိုး ရှင်း ဆုံး ကာဗိုဟိုက်ဒရိတ် မှာ ဂလူးကို့စ် ဂလက်တို့စ် ဖရုပ်တို့စ် စသည့် မို
နို ဆက် က ရိုက် များ ဖြစ် သည်

0.14 Evaluation for dict_sylfreq_bimmfallback_bimmboost50 (closed-test)

[39]:
```
!python ./tools/eval_segmentation.py -r ./data/10k_test.txt -H ./exp_1/
↪dict_sylfreq_bimmfallback_bimmboost50.seg --top-k 10
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                 0.7908
Word Recall                    0.8391
Word F1-score                  0.8142
------------------------------------------------------------
Boundary Precision             0.4829
Boundary Recall                0.5124
Boundary F1-score              0.4972
------------------------------------------------------------
Vocab Precision                0.8078
Vocab Recall                   0.8093
Vocab F1-score                 0.8086
============================================================

Additional Statistics:
Reference words: 117857
Hypothesis words: 125044
Correct words: 98889
Reference vocabulary size: 10840
Hypothesis vocabulary size: 10860
Common vocabulary: 8773

Top Segmentation Errors Analysis
```

```
============================================================
Total errors: 55000

Most Frequent Over-Segmentation Errors (System split where it shouldn't):

Most Frequent Under-Segmentation Errors (System joined what should be separate):

Most Frequent Complex Boundary Errors:
  115 × REF: 'ပါ' → HYP: 'တယ်'
   68 × REF: 'မြန်မာ' → HYP: 'မြန်'
   64 × REF: 'သည်' → HYP: 'အ'
   54 × REF: 'ဖြစ်' → HYP: 'သည်'
   51 × REF: 'နိုင်ငံ' → HYP: 'မာ'
   50 × REF: 'ပေး' → HYP: 'ပါ'
   45 × REF: 'တယ်' → HYP: 'ပါ'
   45 × REF: 'သည်' → HYP: 'ခဲ့'
   44 × REF: 'ချင်' → HYP: 'ပါ'
   44 × REF: 'ကို' → HYP: 'အ'
```

ရလဒ်က အများကြီး တက်လာတာကို တွေ့ရတယ် :)

0.15 4.dict_sylfreq_bimmfallback_bimmboost100 (closed-test)

ဒီတခါ bimmboost ကို 100 ထိ တိုးကြည့်မယ်။

[45]:
```
!time python oppa_word.py \
  --input ./data/10k_test.txt \
  --output "exp_1/dict_sylfreq_bimmfallback_bimmboost100.seg" \
  --postrule-file "./data/rules.txt" \
  --space-remove-mode "my_not_num" \
  --dict "./data/myg2p_mypos.dict" \
  --sylfreq "./data/myMono.freq" \
  --use-bimm-fallback \
  --bimm-boost 100
```

```
real    0m1.074s
user    0m1.066s
sys     0m0.007s
```

[46]:
```
!head ./exp_1/dict_sylfreq_bimmfallback_bimmboost100.seg
```

၁၉၆၂ ခုနှစ် ခန့်မှန်း သန်းခေါင်စာရင်း အရ လူဦး ရေ ၁၁၅၉၃၁ ယောက် ရှိ သည်
လူ တိုင်း တွင် သင့်မြတ် လျော်ကန် စွာ ကန့်သတ် ထား သည့် အလုပ် လုပ်ချိန် အပြင် လစာ
နှင့်တကွ အခါ ကာလ အားလျော်စွာ သတ်မှတ် ထား သည့် အလုပ် အားလပ်ရက် များ ပါဝင် သည့်
အနားယူခွင့် နှင့် အားလပ်ခွင့် ခံစားပိုင်ခွင့် ရှိ သည်
ဤ နည်း ကို စစ်ယူ သော နည်း ဟု ခေါ် သည်

စာပြန်ပွဲ ဆို တာ က အာဂုံဆောင် အလွတ်ကျက် ထား တဲ့ ပိဋကတ်သုံးပုံ စာပေ တွေ ကို စာစစ် သံဃာတော်ကြီး တွေ ရဲ့ ရှေ့မှာ အလွတ် ပြန် ပြီး ရွတ်ပြ ရ တာ ပေါ့
ဒီ မှာ ကျွန်တော့် သက်သေခံကတ် ပါ
၂ ဝ ရာစု မြန်မာ့ သမိုင်း သန်း ဝင်း လှိုင် ၂ ဝ ဝ၉ ခု မေ လ ကံကော်ဝတ်ရည် စာပေ
ကျွန်တော် မျက်မှန် တစ် လက်လုပ် ချင် ပါ တယ်
ကျွန်တော် တို့ က ဒီ အမှု ရဲ့ ကြံရာပါ ကို ဖမ်းမိ ဖို့ ကြိုးစား ခဲ့ တယ်
ကလေး မီးဖွား ဖို့ ခန့်မှန်း ရက် က ဘယ်တော့ ပါ လဲ
အရိုးရှင်းဆုံး ကာဗိုဟိုက်ဒရိတ် မှာ ဂလူးကို့စ် ဂလက်တို့စ် ဖရုပ်တို့စ် စသည့်
မိုနိုဆက်ကရိုက် များ ဖြစ် သည်

0.16 Evaluation for dict_sylfreq_bimmfallback_bimmboost100 (closed-test)

[47]:
```
!python ./tools/eval_segmentation.py -r ./data/10k_test.txt -H ./exp_1/
↪dict_sylfreq_bimmfallback_bimmboost100.seg --top-k 10
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                 0.9028
Word Recall                    0.8805
Word F1-score                  0.8915
------------------------------------------------------------
Boundary Precision             0.6258
Boundary Recall                0.6103
Boundary F1-score              0.6180
------------------------------------------------------------
Vocab Precision                0.8246
Vocab Recall                   0.9204
Vocab F1-score                 0.8699
============================================================

Additional Statistics:
Reference words: 117857
Hypothesis words: 114946
Correct words: 103777
Reference vocabulary size: 10840
Hypothesis vocabulary size: 12099
Common vocabulary: 9977

Top Segmentation Errors Analysis
============================================================
Total errors: 41558

Most Frequent Over-Segmentation Errors (System split where it shouldn't):
```

```
Most Frequent Under-Segmentation Errors (System joined what should be separate):

Most Frequent Complex Boundary Errors:
  154 × REF: 'ပါ' → HYP: 'တယ်'
  100 × REF: 'ဖြစ်' → HYP: 'သည်'
   73 × REF: 'ခဲ့' → HYP: 'သည်'
   59 × REF: 'ကြ' → HYP: 'သည်'
   58 × REF: 'ခဲ့' → HYP: 'တယ်'
   55 × REF: 'ပေး' → HYP: 'ပါ'
   54 × REF: 'များ' → HYP: 'ကို'
   51 × REF: 'ချင်' → HYP: 'ပါ'
   51 × REF: 'သည်' → HYP: 'ခဲ့'
   50 × REF: 'မ' → HYP: 'ရှိ'
```

ရလဒ် ထပ်တက်လာတာကို ရှာဖွေတွေ့ရှိတယ်။
–bimm-boost 50 နဲ့ 100 အကြားမှာ...
Correct words: 98889 ကနေ 103777 ထိ တက်လာတယ်။

0.17 5.dict_sylfreq_arpa_bimmfallback_bimmboost100 (closed-test)

ဒီတခါတော့ LM ပါ သုံးကြည့်မယ်။

[48]:
```
!time python oppa_word.py \
  --input ./data/10k_test.txt \
  --output "exp_1/dict_sylfreq_arpa_bimmfallback_bimmboost100.seg" \
  --postrule-file "./data/rules.txt" \
  --space-remove-mode "my_not_num" \
  --dict "./data/myg2p_mypos.dict" \
  --sylfreq "./data/myMono.freq" \
  --arpa "./data/myMono_clean_syl.arpa" \
  --use-bimm-fallback \
  --bimm-boost 100
```

```
real    0m32.316s
user    0m29.952s
sys     0m2.354s
```

ARPA format LM ထည့်လိုက်တဲ့အတွက် running time က ၃၂ စက္ကန့်တော့ ကြာသွားတယ်။

0.18 Evaluation for dict_sylfreq_arpa_bimmfallback_bimmboost100 (closed-test)

[49]:
```
!python ./tools/eval_segmentation.py -r ./data/10k_test.txt -H ./exp_1/
 ↪dict_sylfreq_arpa_bimmfallback_bimmboost100.seg --top-k 10
```

```
Word Segmentation Evaluation Results
```

```
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                0.8998
Word Recall                   0.8795
Word F1-score                 0.8895
------------------------------------------------------------
Boundary Precision            0.6225
Boundary Recall               0.6085
Boundary F1-score             0.6154
------------------------------------------------------------
Vocab Precision               0.8244
Vocab Recall                  0.9179
Vocab F1-score                0.8686
============================================================

Additional Statistics:
Reference words: 117857
Hypothesis words: 115207
Correct words: 103659
Reference vocabulary size: 10840
Hypothesis vocabulary size: 12070
Common vocabulary: 9950

Top Segmentation Errors Analysis
============================================================
Total errors: 41802

Most Frequent Over-Segmentation Errors (System split where it shouldn't):

Most Frequent Under-Segmentation Errors (System joined what should be separate):

Most Frequent Complex Boundary Errors:
  154 × REF: 'ပါ' → HYP: 'တယ်'
   99 × REF: 'ဖြစ်' → HYP: 'သည်'
   70 × REF: 'ခဲ့' → HYP: 'သည်'
   59 × REF: 'ကြ' → HYP: 'သည်'
   58 × REF: 'ခဲ့' → HYP: 'တယ်'
   55 × REF: 'ပေး' → HYP: 'ပါ'
   54 × REF: 'များ' → HYP: 'ကို'
   51 × REF: 'ချင်' → HYP: 'ပါ'
   50 × REF: 'မ' → HYP: 'ရှိ'
   49 × REF: 'ချင်' → HYP: 'တယ်'
```

LM ထည့်လိုက်တော့မှ

Correct words: 103777 ကနေ 103659 ထိ အနည်းငယ်ကျသွားတာကို တွေ့ရတယ်။

P, R, F-1 score ရလဒ်တွေလည်း အနည်းငယ်လျော့သွားတာကို တွေ့ရတယ်။
နားလည်တာက syllable LM က အရမ်းကြီး အထောက်အပံ့မပြုဘူး ဆိုတဲ့အချက်။

0.19 6.dict_only_bimmfallback_bimmboost150 (closed-test)

ဒီတခါတော့ dictionary တခုတည်းကိုပဲ သုံးပြီး bimmfallback နဲ့ bimmboost ကို 150 ထိ ထားပြီး experiment လုပ်ကြည့်မယ်။

[56]:
```
!time python oppa_word.py \
  --input "./data/10k_test.txt" \
  --output "exp_1/dict_only_bimmfallback_bimmboost150.seg" \
  --postrule-file "./data/rules.txt" \
  --space-remove-mode "my_not_num" \
  --dict "./data/myg2p_mypos.dict" \
  --use-bimm-fallback \
  --bimm-boost 150
```

```
real    0m0.880s
user    0m0.862s
sys     0m0.011s
```

Dictionary ပဲ သုံးတဲ့အတွက် run time ကတော့ အရမ်းမြန်တယ်။

[57]:
```
!head ./exp_1/dict_only_bimmfallback_bimmboost150.seg
```

```
၁၉၆၂ ခုနှစ် ခန့်မှန်း သန်းခေါင်စာရင်း အရ လူဦး ရေ ၁၁၅၉၃၁ ယောက် ရှိ သည်
လူ တိုင်း တွင် သင့်မြတ် လျော်ကန် စွာ ကန့်သတ် ထား သည့် အလုပ် လုပ်ချိန် အပြင် လစာ
နှင့်တကွ အခါ ကာလ အားလျော်စွာ သတ်မှတ် ထား သည့် အလုပ် အားလပ်ရက် များ ပါဝင် သည့်
အနားယူခွင့် နှင့် အားလပ်ခွင့် ခံစားပိုင်ခွင့် ရှိ သည်
ဤ နည်း ကို စစ်ယူ သော နည်း ဟု ခေါ် သည်
စာပြန်ပွဲ ဆို တာ က အာဂုံဆောင် အလွတ်ကျက် ထား တဲ့ ပိဋကတ်သုံးပုံ စာပေ တွေ ကို စာစစ်
သံဃာတော်ကြီး တွေ ရဲ့ ရှေ့မှာ အလွတ် ပြန် ပြီး ရွတ်ပြ ရ တာ ပေါ့
ဒီ မှာ ကျွန်တော့် သက်သေခံကတ် ပါ
၂ ဝ ရာစု မြန်မာ့ သမိုင်း သန်း ဝင်း လှိုင် ၂ ဝ ဝ၉ ခု မေ လ ကံကော်ဝတ်ရည် စာပေ
ကျွန်တော် မျက်မှန် တစ် လက်လုပ် ချင် ပါ တယ်
ကျွန်တော် တို့ က ဒီ အမှု ရဲ့ ကြံရာပါ ကို ဖမ်းမိ ဖို့ ကြိုးစား ခဲ့ တယ်
ကလေး မီးဖွား ဖို့ ခန့်မှန်း ရက် က ဘယ်တော့ ပါ လဲ
အရိုးရှင်းဆုံး ကာဗိုဟိုက်ဒရိတ် မှာ ဂလူးကို့စ် ဂလက်တို့စ် ဖရုပ်တို့စ် စသည့်
မိုနိုဆက်ကရိုက် များ ဖြစ် သည်
```

0.20 Evaluation of dict_only_bimmfallback_bimmboost150 (closed-test)

[58]:
```
!python ./tools/eval_segmentation.py -r ./data/10k_test.txt -H ./exp_1/
↪dict_only_bimmfallback_bimmboost150.seg --top-k 10
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                 0.9034
Word Recall                    0.8807
Word F1-score                  0.8919
------------------------------------------------------------
Boundary Precision             0.6263
Boundary Recall                0.6106
Boundary F1-score              0.6184
------------------------------------------------------------
Vocab Precision                0.8245
Vocab Recall                   0.9208
Vocab F1-score                 0.8700
============================================================

Additional Statistics:
Reference words: 117857
Hypothesis words: 114896
Correct words: 103794
Reference vocabulary size: 10840
Hypothesis vocabulary size: 12107
Common vocabulary: 9982

Top Segmentation Errors Analysis
============================================================
Total errors: 41517

Most Frequent Over-Segmentation Errors (System split where it shouldn't):

Most Frequent Under-Segmentation Errors (System joined what should be separate):

Most Frequent Complex Boundary Errors:
  155 × REF: 'ပါ' → HYP: 'တယ်'
  100 × REF: 'ဖြစ်' → HYP: 'သည်'
   73 × REF: 'ခဲ့' → HYP: 'သည်'
   59 × REF: 'ကြ' → HYP: 'သည်'
   58 × REF: 'ခဲ့' → HYP: 'တယ်'
   55 × REF: 'ပေး' → HYP: 'ပါ'
   54 × REF: 'များ' → HYP: 'ကို'
   52 × REF: 'ချင်' → HYP: 'ပါ'
   52 × REF: 'သည်' → HYP: 'ခဲ့'
   50 × REF: 'မ' → HYP: 'ရှိ'
```

အခုချိန်ထိ ရလဒ်အများဆုံး …

0.21 7. bimm-boost Up to 200 (closed-test)

–bimm-boost 200 ထိ တိုးပြီး dictionary only နဲ့ပဲ ထပ်စမ်းကြည့်ခဲ့တယ်။

[59]:
```
!time python oppa_word.py \
  --input "./data/10k_test.txt" \
  --output "exp_1/dict_only_bimmfallback_bimmboost200.seg" \
  --postrule-file "./data/rules.txt" \
  --space-remove-mode "my_not_num" \
  --dict "./data/myg2p_mypos.dict" \
  --use-bimm-fallback \
  --bimm-boost 200
```

```
real    0m0.881s
user    0m0.875s
sys     0m0.006s
```

[60]:
```
!python ./tools/eval_segmentation.py -r ./data/10k_test.txt -H ./exp_1/
 ↪dict_only_bimmfallback_bimmboost200.seg --top-k 10
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                 0.9034
Word Recall                    0.8807
Word F1-score                  0.8919
------------------------------------------------------------
Boundary Precision             0.6263
Boundary Recall                0.6106
Boundary F1-score              0.6184
------------------------------------------------------------
Vocab Precision                0.8245
Vocab Recall                   0.9208
Vocab F1-score                 0.8700
============================================================

Additional Statistics:
Reference words: 117857
Hypothesis words: 114896
Correct words: 103794
Reference vocabulary size: 10840
Hypothesis vocabulary size: 12107
Common vocabulary: 9982
```

```
Top Segmentation Errors Analysis
============================================================
Total errors: 41517

Most Frequent Over-Segmentation Errors (System split where it shouldn't):

Most Frequent Under-Segmentation Errors (System joined what should be separate):

Most Frequent Complex Boundary Errors:
  155 × REF: 'ပါ' → HYP: 'တယ်'
  100 × REF: 'ဖြစ်' → HYP: 'သည်'
   73 × REF: 'ခဲ့' → HYP: 'သည်'
   59 × REF: 'ကြ' → HYP: 'သည်'
   58 × REF: 'ခဲ့' → HYP: 'တယ်'
   55 × REF: 'ပေး' → HYP: 'ပါ'
   54 × REF: 'များ' → HYP: 'ကို'
   52 × REF: 'ချင်' → HYP: 'ပါ'
   52 × REF: 'သည်' → HYP: 'ခဲ့'
   50 × REF: 'မ' → HYP: 'ရှိ'
```

ရလဒ်က ဒီထက် ဆက်မတက်နိုင်တော့တာကို တွေ့ရတယ်။

အဲဒါကြောင့် open test ကို လက်ရှိ နောက်ဆုံး setting နဲ့ပဲ စမ်းယုံနဲ့ လုံလောက်တယ်လို့ ယူဆခဲ့တယ်။
(တကယ်က ဒီ notebook နဲ့ demo လုပ်မပြခင် experiment အများကြီး လုပ်ထားပြီးသားပါ။)
open-test ဆိုရင် word segmentation ဖြတ်ရတာ ပိုခက်တာမို့လို့ ...

0.22 1. dict_only_bimmfallback_bimmboost150 (open-test)

ဒီတခါတော့ open test စာကြောင်းရေ တစ်ထောင်နဲ့ စမ်းမယ်။ ရှေ့က run ထားခဲ့တဲ့ baseline approach တွေနဲ့ နှိုင်းယှဉ်ကြည့်မယ်။

[61]:
```
!time python oppa_word.py \
  --input "./data/otest.1k.word" \
  --output "exp_1/dict_only_bimmfallback_bimmboost150_otest.seg" \
  --postrule-file "./data/rules.txt" \
  --space-remove-mode "my_not_num" \
  --dict "./data/myg2p_mypos.dict" \
  --use-bimm-fallback \
  --bimm-boost 150
```

```
real    0m0.133s
user    0m0.128s
sys     0m0.004s
```

[62]:
```
!head ./exp_1/dict_only_bimmfallback_bimmboost150_otest.seg
```

တစ် ကိုက် ကို ဝမ် ခုနှစ်ထောင် ပါ ။
မနှစ် က သူ ကျွန်မ ကို သင်ပေး တယ် ။
ကျွန်တော့် ခုံ သွား ရှာ မလို့ ။
အတန်း စတာ ကြာ ပြီ လား ။
ဆေး နည်းနည်း စား လိုက် ၊ သုံး လေး ရက် လောက် အနားယူ လိုက် ရင် ပျောက် သွား မှာ ပါ ။
အေးချမ်း မှု နဲ့ စည်းကမ်း ကို တည်မြဲ အောင် ထိန်းသိမ်း သည် ။
ဇွန်း ကို လိုအပ် တယ် ။
ဘွဲ့ရ ရင် ဘာ လုပ် မလို့ လဲ ။
ကျွန်တော် ချောင်းဆိုး ခြင်း အတွက် တစ်ခုခု လို ချင် တယ် ။
အသီးအနှံ တို့ မှ လွဲလျှင် လူ တို့ ၏ အဓိက အစားအစာ မှာ ငါး ဖြစ် သည် ။

[64]:
```
!tail ./exp_1/dict_only_bimmfallback_bimmboost150_otest.seg
```

အိုးခွက်ပန်းကန် တွေ သိပ် မ ရှိ လို့ ထမင်းဟင်း ချက် ရ တာ အဆင်မပြေ ဘူး ။
စိတ်ဝင်စား ဖို့ ကောင်း တယ် ။
ဒီ ဆေး ကို တဝက် စီ ခွဲ ပေး ပါ ။
ရောင်း ကောင်း လား ။
ဆရာ ဒီ သွား က ခဏခဏ နာ နေ တယ် ။
အခု ဘာ လုပ် နေ လဲ ။
ဇူ လိုင် ၁၄ ရက် မှာ ဘန်ကောက် ကို သွား မယ့် US 123 မှာ ပါ ။ ဟုတ် လား ။
ကား မှ ကားဘီး ကို ဖြုတ် လိုက် သည် ။
ကျွန်တော် သိ ပါရစေ ။
ဘူတာရုံ က အလွန်တရာ ပြည့်ကျပ် နေ သည် ။

0.23 Evaluation of dict_only_bimmfallback_bimmboost150 (open-test)

[66]:
```
!python ./tools/eval_segmentation.py -r ./data/otest.1k.word -H ./exp_1/
↪dict_only_bimmfallback_bimmboost150_otest.seg --top-k 10
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                 0.9094
Word Recall                    0.8893
Word F1-score                  0.8992
------------------------------------------------------------
Boundary Precision             0.6116
Boundary Recall                0.5981
Boundary F1-score              0.6048
------------------------------------------------------------
Vocab Precision                0.8183
Vocab Recall                   0.9158
```

```
Vocab F1-score                0.8643
============================================================

Additional Statistics:
Reference words: 13468
Hypothesis words: 13170
Correct words: 11977
Reference vocabulary size: 2709
Hypothesis vocabulary size: 3032
Common vocabulary: 2481

Top Segmentation Errors Analysis
============================================================
Total errors: 4966

Most Frequent Over-Segmentation Errors (System split where it shouldn't):

Most Frequent Under-Segmentation Errors (System joined what should be separate):

Most Frequent Complex Boundary Errors:
   50 × REF: 'တယ်' → HYP: '။'
   49 × REF: 'သည်' → HYP: '။'
   39 × REF: 'ပါ' → HYP: '။'
   20 × REF: 'ပါ' → HYP: 'တယ်'
   19 × REF: '။' → HYP: 'သည်'
   19 × REF: 'မယ်' → HYP: '။'
   17 × REF: 'လား' → HYP: '။'
   17 × REF: 'ဖြစ်' → HYP: 'သည်'
   13 × REF: 'ဘူး' → HYP: '။'
   12 × REF: 'လဲ' → HYP: '။'
```

0.24 Summary

Typical Score Ranges:
- Dictionary scores: ~10.0 (from –dict-weight) - Syllable scores: Typically 0-5 (log frequency based)
- LM scores: Negative values (log probabilities), often -20 to 0

Impact Analysis:
- At boost=50: Already dominates other scores (10 + -20 + 50 = 40) - At boost=150: (10 + -20 + 150 = 140) - completely overrides other factors - Beyond 150: No improvement because BiMM paths already always win

oppaWord ရဲ့ run time ကတော့ တအားမြန်တယ်။
ခုချိန်က စပြီး ရှေ့က run ခဲ့တဲ့ baseline တွေနဲ့ oppaWord ရဲ့ ရလဒ်တွေကို နှိုင်းယှဉ်ကြည့်မယ်။

0.25 Word Metrics (Closed Test)

Model	Precision	Recall	F1-score
myWord	0.8235	0.8958	0.8581
CRF	0.6928	0.6961	0.6944
LSTM	0.8970	0.9210	**0.9088**
oppaWord	0.9034	0.8807	0.8919

0.26 Boundary Metrics (Closed Test)

Model	Precision	Recall	F1-score
myWord	0.4695	0.5107	0.4893
CRF	0.3186	0.3201	0.3194
LSTM	0.6119	0.6283	**0.6200**
oppaWord	0.6263	0.6106	0.6184

0.27 Vocab Metrics (Closed Test)

Model	Precision	Recall	F1-score
myWord	0.8953	0.6456	0.7502
CRF	0.3256	0.4674	0.3838
LSTM	0.7265	0.7093	0.7178
oppaWord	0.8245	0.9208	**0.8700**

0.28 Additional Evaluation Statistics (Closed Test)

Metric	**myWord**	**CRF**	**LSTM**	**oppaWord**
Reference words	117857	117857	117857	117857
Hypothesis words	128206	118425	121013	114896
Correct words	105579	82042	108546	103794
Reference vocab size	10840	10840	10840	10840
Hypothesis vocab size	7816	15563	10584	12107
Common vocabulary	6998	5067	7689	9982
Total segmentation errors	57570	74443	42500	41517

Open Test နဲ့ word segmentation ရလဒ်တွေကို နှိုင်းယှဉ်ကြည့်မယ်။

0.29 Word Metrics (Open Test)

Model	Precision	Recall	F1-score
myWord	0.8489	0.9121	0.8793
CRF	0.7019	0.7029	0.7024
LSTM	0.9055	0.9266	**0.9159**

Model	Precision	Recall	F1-score
oppaWord	0.9094	0.8893	0.8992

0.30 Boundary Metrics (Open Test)

Model	Precision	Recall	F1-score
myWord	0.4769	0.5124	0.4940
CRF	0.3210	0.3215	0.3213
LSTM	0.6025	0.6165	**0.6094**
oppaWord	0.6116	0.5981	0.6048

0.31 Vocab Metrics (Open Test)

Model	Precision	Recall	F1-score
myWord	0.8911	0.7645	0.8230
CRF	0.4646	0.5482	0.5030
LSTM	0.8091	0.8169	0.8130
oppaWord	0.8183	0.9158	**0.8643**

0.32 Additional Evaluation Statistics (Open Test)

Metric	**myWord**	**CRF**	**LSTM**	**oppaWord**
Reference words	13468	13468	13468	13468
Hypothesis words	14471	13487	13782	13170
Correct words	12284	9467	12479	11977
Reference vocab size	2709	2709	2709	2709
Hypothesis vocab size	2324	3196	2735	3032
Common vocabulary	2071	1485	2213	2481
Total segmentation errors	6568	8545	5042	4966

0.33 Conclusion or Findings

- ရလဒ်တွေအားလုံးကနေ နားလည်တာက oppaWord run time က super fast ပါပဲ။
- အထူးသဖြင့် myWord ထက် အများကြီး မြန်တယ်။
- Word Segmentation performance မှာတော့ LSTM ကို နီးပါးရတယ်။
- oppaWord ရဲ့ အားသာချက်က training data ကိုသုံးပြီး training လုပ်စရာ မလိုဘူး။
- သေချာတာက dictionary ကို ထပ်ဖြည့်သွားရင် ရလဒ်က ပိုတက်လာလိမ့်မယ်။
- ဒီ experiment ကနေ ထပ်သိရတာက ငါတို့ monolingual ဒေတာကို ထပ်အားဖြည့်မှ ရလိမ့်မယ်။ လောလောဆယ် တစ်သန်းကျော်နဲ့ syllable LM က လက်ရှိ oppaWord approach နဲ့ဆိုရင်တော့ struggle လုပ်နေရတုန်းပဲ။ ဖြစ်နိုင်ရင်တော့ word segmentation ဖြတ်ထားတဲ့ ဒေတာနဲ့ LM အကြီးကြီး ဆောက်နိုင်ဖို့ လိုအပ်တယ်။

- ပြီးတော့ bidirectional maximum matching ရဲ့ power ကိုလည်း ဒီ experiment ကနေ ထပ် confirm လုပ်လို့ ရတယ်။
- oppaWord ရဲ့ proposal မှာ –use-bimm-fallback နဲ့ –bimm-boost BIMM_BOOST ကို သုံးပြထားတယ်။
- input file ကိုလည်း oppaWord က –space-remove-mode သုံးရင် space ရှင်းပေးတာမို့ user အနေနဲ့ ပုံမှန်မြန်မာစာဖိုင်ကို ဒီအတိုင်း input လုပ်သုံးလို့ ရတယ်။
- rule ဖိုင်နဲ့လည်း segmentation အမှားတွေကို ပြင်လို့ ရတာက လက်တွေ့ word segmentation အလုပ်တွေအတွက် အသုံးဝင်လိမ့်မယ်
- ပြီးတော့ educational purpose အနေနဲ့ DAG visualization ကိုလည်း graph အနေနဲ့ ထုတ်ပြပေးနိုင်တယ်

0.34 To Do

- Name dictionary ကို ဖြည့်ပြီး experiment လုပ်ရန်
- အားလုံးသုံးလို့ ရအောင် oppaWord ကို အမြန်ဆုံး release လုပ်ရန်
- myNLP မှာ oppaWord ကို ဖြည့်ရန်

[ ]: