baselines

August 3, 2025

0.1 Word Segmentation Baselines

ဒီ Notebook မှာ လက်ရှိ LU Lab. ရဲ့ myTokenizers က support လုပ်တဲ့ word segmentation method သုံးမျိုးကို သုံးပြီး baseline ထုတ်ကြည့်မယ်။

For Internship-3 of LU Lab Students.
Prepared by Ye, LU Lab., Myanmar.
Date: 26 July 2025

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

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

[1]:
```
!python --version
```

```
Python 3.10.8
```

[5]:
```
!echo $PYTHONPATH
```

```
/home/ye/miniforge3/envs/myTokenize/bin/
```

[6]:
```
!conda list
```

```
# packages in environment at /home/ye/miniforge3/envs/myTokenize:
#
# Name                    Version                   Build  Channel
_libgcc_mutex             0.1                 conda_forge    conda-forge
_openmp_mutex             4.5                       2_gnu    conda-forge
absl-py                   2.3.1                     pypi_0    pypi
astunparse                1.6.3                     pypi_0    pypi
bzip2                     1.0.8                h4bc722e_7    conda-forge
ca-certificates           2025.7.14            hbd8a1cb_0    conda-forge
cachetools                5.5.2                     pypi_0    pypi
certifi                   2025.7.14                 pypi_0    pypi
charset-normalizer        3.4.2                     pypi_0    pypi
contourpy                 1.3.2                     pypi_0    pypi
cycler                    0.12.1                    pypi_0    pypi
flatbuffers               25.2.10                   pypi_0    pypi
fonttools                 4.59.0                    pypi_0    pypi
gast                      0.4.0                     pypi_0    pypi
gitdb                     4.0.12                    pypi_0    pypi
```

```
gitpython                 3.1.41                  pypi_0    pypi
google-auth               2.40.3                  pypi_0    pypi
google-auth-oauthlib      1.0.0                   pypi_0    pypi
google-pasta              0.2.0                   pypi_0    pypi
grpcio                    1.74.0                  pypi_0    pypi
h5py                      3.14.0                  pypi_0    pypi
icu                       75.1                 he02047a_0    conda-forge
idna                      3.10                    pypi_0    pypi
keras                     2.13.1                  pypi_0    pypi
kiwisolver                1.4.8                   pypi_0    pypi
ld_impl_linux-64          2.44                 h1423503_1    conda-forge
libclang                  18.1.1                  pypi_0    pypi
libffi                    3.4.6                h2dba641_1    conda-forge
libgcc                    15.1.0               h767d61c_3    conda-forge
libgcc-ng                 15.1.0               h69a702a_3    conda-forge
libgomp                   15.1.0               h767d61c_3    conda-forge
liblzma                   5.8.1                hb9d3cd8_2    conda-forge
liblzma-devel             5.8.1                hb9d3cd8_2    conda-forge
libnsl                    2.0.1                hb9d3cd8_1    conda-forge
libsqlite                 3.50.3               hee844dc_1    conda-forge
libstdcxx                 15.1.0               h8f9b012_3    conda-forge
libstdcxx-ng              15.1.0               h4852527_3    conda-forge
libuuid                   2.38.1               h0b41bf4_0    conda-forge
libzlib                   1.3.1                hb9d3cd8_2    conda-forge
markdown                  3.8.2                   pypi_0    pypi
markupsafe                3.0.2                   pypi_0    pypi
matplotlib                3.7.4                   pypi_0    pypi
ml-dtypes                 0.2.0                   pypi_0    pypi
mytokenize                0.1.1                   pypi_0    pypi
ncurses                   6.5                  h2d0b736_3    conda-forge
numpy                     1.24.3                  pypi_0    pypi
oauthlib                  3.3.1                   pypi_0    pypi
openssl                   3.5.1                h7b32b05_0    conda-forge
opt-einsum                3.4.0                   pypi_0    pypi
packaging                 25.0                    pypi_0    pypi
pillow                    11.3.0                  pypi_0    pypi
pip                       25.1.1               pyh8b19718_0    conda-forge
protobuf                  4.25.8                  pypi_0    pypi
pyasn1                    0.6.1                   pypi_0    pypi
pyasn1-modules            0.4.2                   pypi_0    pypi
pyparsing                 3.2.3                   pypi_0    pypi
python                    3.10.8          h4a9ceb5_0_cpython    conda-forge
python-crfsuite           0.9.9                   pypi_0    pypi
python-dateutil           2.9.0.post0             pypi_0    pypi
readline                  8.2                  h8c095d6_2    conda-forge
requests                  2.32.4                  pypi_0    pypi
requests-oauthlib         2.0.0                   pypi_0    pypi
rsa                       4.9.1                   pypi_0    pypi
```

```
sentencepiece              0.2.0                    pypi_0    pypi
setuptools                 80.9.0               pyhff2d567_0    conda-forge
six                        1.17.0                   pypi_0    pypi
smmap                      5.0.2                    pypi_0    pypi
tensorboard                2.13.0                   pypi_0    pypi
tensorboard-data-server    0.7.2                    pypi_0    pypi
tensorflow                 2.13.0                   pypi_0    pypi
tensorflow-addons          0.21.0                   pypi_0    pypi
tensorflow-estimator       2.13.0                   pypi_0    pypi
tensorflow-io-gcs-filesystem 0.37.1                    pypi_0    pypi
termcolor                  3.1.0                    pypi_0    pypi
tk                         8.6.13          noxft_hd72426e_102    conda-forge
typeguard                  2.13.3                   pypi_0    pypi
typing-extensions          4.5.0                    pypi_0    pypi
tzdata                     2025b                h78e105d_0    conda-forge
urllib3                    2.5.0                    pypi_0    pypi
werkzeug                   3.1.3                    pypi_0    pypi
wheel                      0.45.1              pyhd8ed1ab_1    conda-forge
wrapt                      1.14.1                   pypi_0    pypi
xz                         5.8.1                hbcc6ac9_2    conda-forge
xz-gpl-tools               5.8.1                hbcc6ac9_2    conda-forge
xz-tools                   5.8.1                hb9d3cd8_2    conda-forge
```

[1]:
```
import tensorflow
```

```
2025-07-26 01:30:22.735656: I tensorflow/core/util/port.cc:110] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-07-26 01:30:22.736825: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 01:30:22.757513: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 01:30:22.757927: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
2025-07-26 01:30:23.157676: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT
```

0.2 Testing Syllable Segmentation

အရင်ဆုံး Library က အလုပ် လုပ်သလားဆိုတာကို စမ်းကြည့်တာပါ။

[48]:
```
from myTokenize import SyllableTokenizer
```

```
tokenizer = SyllableTokenizer()
syllables = tokenizer.tokenize("ဗမာစကားပြောတတ်လား။")
print(syllables)
```

```
['ဗ', 'မာ', 'စ', 'ကား', 'ပြော', 'တတ်', 'လား', '။']
```

0.3 Preparing Open-test Data

မြန်မာစာအတွက်က manual word segmentation ဖြတ်ထားပြီးသား data ကလည်း မရှိသလောက်ပဲဆိုတော့... myPOS (version 3.0) ရဲ့ tag မပါတဲ့ open-test dataset ကိုပဲ word segmentation အတွက်လည်း သုံးမယ်။

[3]:
```
!mkdir data
```

[4]:
```
cd data
```

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

[5]:
```
!wget https://raw.githubusercontent.com/ye-kyaw-thu/myPOS/refs/heads/master/
↪corpus-ver-3.0/corpus/otest.1k.txt
```

```
--2025-07-26 01:34:46--  https://raw.githubusercontent.com/ye-kyaw-
thu/myPOS/refs/heads/master/corpus-ver-3.0/corpus/otest.1k.txt
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.111.133, 185.199.109.133, 185.199.108.133, …
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.111.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 229758 (224K) [text/plain]
Saving to: ‘otest.1k.txt’

otest.1k.txt        100%[===================>] 224.37K   402KB/s    in 0.6s

2025-07-26 01:34:48 (402 KB/s) - ‘otest.1k.txt’ saved [229758/229758]
```

data မှာက POS Tag တွေ ပါနေသေးတယ်။

[6]:
```
!head otest.1k.txt
```

```
တစ်/tn ကိုက်/n ကို/ppm ဝမ်/n ခုနှစ်ထောင်/tn ပါ/part ။/punc
မနှစ်/n က/ppm သူ/pron ကျွန်မ/pron ကို/ppm သင်/v ပေး/part တယ်/ppm ။/punc
ကျွန်တော့်/pron ခုံ/n သွား/v ရှာ/v မလို့/part ။/punc
အတန်း/n စ/v တာ/part ကြာ/v ပြီ/ppm လား/part ။/punc
ဆေး/n နည်းနည်း/adv စား/v လိုက်/part ၊/punc သုံး/tn လေး/tn ရက်/n လောက်/part
အနားယူ/v လိုက်/part ရင်/conj ပျောက်/v သွား/part မှာ/ppm ပါ/part ။/punc
အေးချမ်း/v၊ မှု/part နဲ့/conj စည်းကမ်း/n ကို/ppm တည်မြဲ/v အောင်/part ထိန်းသိမ်း/v
သည်/ppm ။/punc
ဇွန်း/n ကို/ppm လိုအပ်/v တယ်/ppm ။/punc
```

ဘွဲ့/n ရ/v ရင်/conj ဘာ/n လုပ်/v မ/part လို့/part လဲ/part ။/punc
ကျွန်တော်/pron ချောင်းဆိုး/v ခြင်း/part အတွက်/ppm တစ်/tn ခု/part ခု/part လို/v
ချင်/part တယ်/ppm ။/punc
အသီးအနှံ/n တို့/part မှ/ppm လွဲ/v လျှင်/conj လူ/n တို့/part ၏/ppm အဓိက/n
အစားအစာ/n မှာ/ppm ငါး/n ဖြစ်/v သည်/ppm ။/punc

0.4 Removing POS Tags

Tag တွေ၊ word တွေချည်းပဲ ဆွဲထုတ်ပေးတဲ့ perl script ကို ကိုယ့်စက်ထဲကို download လုပ်ယူမယ်။

[7]:
```
!wget https://raw.githubusercontent.com/ye-kyaw-thu/myPOS/refs/heads/master/
↪corpus-draft-ver-1.0/mk-wordtag.pl
```

```
--2025-07-26 01:37:52--  https://raw.githubusercontent.com/ye-kyaw-
thu/myPOS/refs/heads/master/corpus-draft-ver-1.0/mk-wordtag.pl
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.111.133, 185.199.109.133, 185.199.110.133, …
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.111.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 3967 (3.9K) [text/plain]
Saving to: ‘mk-wordtag.pl’

mk-wordtag.pl       100%[===================>]   3.87K  --.-KB/s    in 0s

2025-07-26 01:37:53 (49.5 MB/s) - ‘mk-wordtag.pl’ saved [3967/3967]
```

Perl code ကို လေ့လာကြည့်ပါ။
ဖိုင်နာမည်က mk-wordtag.pl ပါ။

```
#!/usr/bin/perl
use warnings;
use utf8;

#last updated: 16 May 2017
#written by Ye, AI Lab.,
#Okayama Prefectural University, Japan
#How to run: perl mk-wordtag.pl <input-file-name> <delimeter> <w|t|wt|cw|c>
#Here,
# w = print word only (i.e. without POS tags),
# t = print tag only
# wt = print word/tag
# cw = print words including compound words,
# lcw = list compound words,
# c = print sentence that contain tagging error of "word/"
#
# How to run:
```

```
# e.g. ./mk-wordtag.pl ./kh-pos.all.f2.utf8 "\/" w | less -r
# e.g ./mk-wordtag.pl ./kh-pos.all.f2.utf8 "\/" t
# e.g ./mk-wordtag.pl ./kh-pos.all.f2.utf8 "\/" wt

binmode STDIN,  ":utf8";
binmode STDOUT, ":utf8";

my $TagMarker=$ARGV[1]; # give command line parameter such as "\|", "\/" ...
my $word_or_tag=$ARGV[2];

open (my $inputFILE,"<:encoding(utf8)", $ARGV[0]) or die "Couldn't open input file $ARGV[0]!,

my $one_token; my $tmpLine=""; my $tmpLine2="";

   while($line = <$inputFILE>)
   {
      if ($line!~/^$/)
      {
         chomp ($line);
         my $originalLine = $line;
         #print $line, "\n";

         $line =~ s/\s+/ /g;
         $line =~ s/^\s+|\s+$//g;
         if ($word_or_tag eq "w" || $word_or_tag eq "t" || $word_or_tag eq "wt" || $word_or_tag
         {
            $line =~ s/\|/ /g;
         }

         my @token = split('\s', $line);
         #print "\@tokens:\n"."@token\n";
         foreach $one_token(@token)
         {
            #print "one_token: $one_token\n";
            my ($text, $tag) = split(/$TagMarker/, $one_token);
            if($word_or_tag eq "w")
            {
               $tmpLine = $tmpLine.$text." ";
            }elsif($word_or_tag eq "t")
            {
               $tmpLine = $tmpLine.$tag." ";
            }elsif($word_or_tag eq "wt" || $word_or_tag eq "c")
            {
               $tmpLine = $tmpLine.$text." ";
               $tmpLine2 = $tmpLine2.$tag." ";
            }elsif($word_or_tag eq "lcw" || $word_or_tag eq "cw")
            {
               if($one_token =~ m/\|/g)
```

```
        {
           my @ptoken = split('\|', $one_token); my $combined_cword;
              foreach my $cword(@ptoken)
              {
                 $cword =~ s/\/[a-zA-Z].*//g;
                 $combined_cword = $combined_cword.$cword;
              }
                 if ($word_or_tag eq "lcw")
                 {
                    print "$one_token\t$combined_cword\n"; #for lcw option;
                 }
            $tmpLine = $tmpLine.$combined_cword." "; #for cw option
         }elsif($one_token !~ m/\|/g)
         {

            $tmpLine = $tmpLine.$text." ";
         }
      }
   }
      #chomp($tmpLine);
      if ($word_or_tag eq "w" || $word_or_tag eq "t" || $word_or_tag eq "cw")
      {
         $tmpLine =~ s/^\s+|\s+$//g;
         print $tmpLine."\n";
      }elsif ($word_or_tag eq "wt")
      {

         $tmpLine =~ s/^\s+|\s+$//g;
         $tmpLine2 =~ s/^\s+|\s+$//g;
         print $tmpLine."\n"; print $tmpLine2."\n";
      }elsif ($word_or_tag eq "c")
      {
         $tmpLine =~ s/^\s+|\s+$//g;
         $tmpLine =~ s/\s+/ /g;
         $tmpLine2 =~ s/^\s+|\s+$//g;
         $tmpLine2 =~ s/\s+/ /g;
         my $word_count = split / /,$tmpLine;
         my $tag_count = split / /,$tmpLine2;

         if ($word_count != $tag_count)
         {
            print "$originalLine\n";
            print "$word_count: $tag_count\n";
            print $tmpLine."\n"; print $tmpLine2."\n";
         }

       }elsif ($word_or_tag eq "lcw")
       {
```

```
            # print $tmpLine2;
        }
                $tmpLine = ""; $tmpLine2 = "";
    }
}

close($inputFILE);
```

word တွေပဲ ဆွဲထုတ်ယူပြီး ဖိုင်အသစ်အနေနဲ့ သိမ်းမယ်...

[9]:
```
!perl ./mk-wordtag.pl ./otest.1k.txt "\/" w > ./otest.1k.word
```

[10]:
```
!head ./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 မှူ 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 ။
```

0.5 Preprocessing

myTokenize ကို input လုပ်တဲ့အခါမှာ space တွေအကုန် ဖြုတ်ပေးထားရတာမို့ အဲဒီအတွက် အောက်ပါ python code ကို သုံးမယ်။ ပရိုဂရမ် နာမည်က smart_space_remover.py ပါ။

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

"""
written by Ye Kyaw Thu, LU Lab., Myanmar
last updated: 25 July 2025
smart_space_remover.py: Remove spaces intelligently for Myanmar text segmentation.

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_not_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()
```

space တွေကို ဖြုတ်မယ်။ –mode ကို my_not_num ထားမယ်။
my_not_num နဲ့ ဆိုရင် မြန်မာ နံပါတ်တွေကို မထိဘူး။ ပြီးတော့ မြန်မာစာ မဟုတ်တဲ့ တခြားဘာသာစကားတွေ ဥပမာ အင်္ဂလိပ်စာ၊ ဂျပန်စာ၊ ထိုင်းစာ တို့ကို space မဖြုတ်ဘူး။

[17]:
```
!python ./smart_space_remover.py --help
```

```
usage: smart_space_remover.py [-h] --mode {all,my,my_not_num} [--input INPUT]
                              [--output OUTPUT]

Smart Myanmar space remover

options:
  -h, --help            show this help message and exit
```

```
  --mode {all,my,my_not_num}
                        Mode: 'all', 'my', or 'my_not_num'
  --input INPUT         Input file (default: stdin)
  --output OUTPUT       Output file (default: stdout)
```

[18]:
```
!python ./smart_space_remover.py --input ./otest.1k.word --output ./otest.1k.
↪word.input --mode my_not_num
```

space ဖြုတ်ပြီး ထွက်လာတဲ့ ဖိုင်ကို **confirm** လုပ်ရအောင်

[19]:
```
!pwd
```

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

[20]:
```
!head ./otest.1k.word.input
```

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

[21]:
```
!tail ./otest.1k.word.input
```

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

0.6 Coding

တကယ်တမ်း လက်တွေ့ word segmentation လုပ်ဖို့က python ပရိဂရမ် တစ်ပုဒ်ရေးထားလိုက်တာက ပိုအဆင်ပြေတာမို့ အောက်ပါ code ကို ရေးခဲ့တယ်။

[22]:
```
cd ..
```

```
/home/ye/exp/myTokenizer
"""
Written by Ye Kyaw Thu., LU Lab., Myanmar.
Last updated: 26 July 2025.
Usage:
time python ./baseline_segmenter.py --input ./data/otest.1k.word.input --output ./output/otest
time python ./baseline_segmenter.py --input ./data/otest.1k.word.input --output ./output/otest
time python ./baseline_segmenter.py --input ./data/otest.1k.word.input --output ./output/otest
"""

import sys
import argparse
import tensorflow
from myTokenize import (
    WordTokenizer,
    SentenceTokenizer,
    SyllableTokenizer,
    BPETokenizer,
    UnigramTokenizer,
    PhraseTokenizer
)

def get_tokenizer(method):
    """Return the appropriate tokenizer based on the selected method"""
    method = method.lower()  # Normalize to lowercase

    if method == 'crf':
        return WordTokenizer(engine="CRF")
    elif method == 'myword':
        return WordTokenizer(engine="myWord")  # Note the exact capitalization
    elif method == 'lstm':
        return WordTokenizer(engine="LSTM")
    elif method == 'sentence':
        return SentenceTokenizer()
    elif method == 'syllable':
        return SyllableTokenizer()
    elif method == 'bpe':
        return BPETokenizer()
    elif method == 'unigram':
        return UnigramTokenizer()
    elif method == 'phrase':
        return PhraseTokenizer()
    else:
        raise ValueError(f"Unknown segmentation method: {method}. Available methods: crf, mywo

def tokenize_text(tokenizer, text):
    """Tokenize text and return as space-separated string"""
```

```
    if not text.strip():
        return text  # return empty lines as-is

    tokens = tokenizer.tokenize(text.strip())

    # For all tokenizers except sentence, join with spaces
    if not isinstance(tokenizer, SentenceTokenizer):
        return ' '.join(tokens)
    # For sentence tokenizer, join sentences with newlines
    return '\n'.join(tokens)

def process_stream(tokenizer, input_stream, output_stream):
    """Process input stream and write to output stream"""
    for line in input_stream:
        tokenized = tokenize_text(tokenizer, line)
        output_stream.write(tokenized + '\n')

def main():
    parser = argparse.ArgumentParser(
        description="Myanmar Text Segmenter",
        formatter_class=argparse.ArgumentDefaultsHelpFormatter
    )
    parser.add_argument(
        '-m', '--method',
        choices=['crf', 'myword', 'lstm', 'sentence', 'syllable', 'bpe', 'unigram', 'phrase'],
        default='crf',
        help='Segmentation method to use (crf, myword, lstm, sentence, syllable, bpe, unigram,
    )
    parser.add_argument(
        '-i', '--input',
        type=str,
        help='Input file path (default: stdin)'
    )
    parser.add_argument(
        '-o', '--output',
        type=str,
        help='Output file path (default: stdout)'
    )

    args = parser.parse_args()

    # Get tokenizer
    try:
        tokenizer = get_tokenizer(args.method)
    except ValueError as e:
        print(f"Error: {e}", file=sys.stderr)
        sys.exit(1)
```

```
    # Setup input and output streams
    input_stream = sys.stdin
    if args.input:
        try:
            input_stream = open(args.input, 'r', encoding='utf-8')
        except IOError as e:
            print(f"Error opening input file: {e}", file=sys.stderr)
            sys.exit(1)

    output_stream = sys.stdout
    if args.output:
        try:
            output_stream = open(args.output, 'w', encoding='utf-8')
        except IOError as e:
            print(f"Error opening output file: {e}", file=sys.stderr)
            if input_stream != sys.stdin:
                input_stream.close()
            sys.exit(1)

    # Process the text
    try:
        process_stream(tokenizer, input_stream, output_stream)
    except Exception as e:
        print(f"Error during processing: {e}", file=sys.stderr)
        sys.exit(1)
    finally:
        # Close files if they were opened
        if args.input and input_stream != sys.stdin:
            input_stream.close()
        if args.output and output_stream != sys.stdout:
            output_stream.close()

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

0.7 Word Segmentation with myWord

အရင်ဆုံး myWord နဲ့ စာလုံးဖြတ်မယ်။
myWord ရဲ့ word segmentation ဖြတ်ပုံ အသေးစိတ်က GitHub myWord မှာ ဝင်လေ့လာပါ။
Link: https://github.com/ye-kyaw-thu/myWord

[24]:
```
!mkdir output
```

[26]:
```
!time python ./baseline_segmenter.py --input ./data/otest.1k.word.input␣
↪--output ./output/otest.myword -m myword
```

```
2025-07-26 03:07:32.687264: I tensorflow/core/util/port.cc:110] oneDNN custom
operations are on. You may see slightly different numerical results due to
```

```
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-07-26 03:07:32.688214: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 03:07:32.708666: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 03:07:32.708914: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
2025-07-26 03:07:33.091311: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT

real    8m58.617s
user    8m0.665s
sys     1m0.425s
```

[27]:
```
!head ./output/otest.myword
```

```
တစ် ကိုက် ကို ဝမ် ခု နှစ် ထောင် ပါ ။
မ နှစ် က သူ ကျွန်မ ကို သင် ပေး တယ် ။
ကျွန်တော့် ခုံ သွား ရှာ မ လို့ ။
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 ။
```

[37]:
```
!tail ./output/otest.myword
```

```
အိုးခွက်ပန်းကန် တွေ သိပ် မ ရှိ လို့ ထမင်း ဟင်း 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 ။
ဇူလိုင် ၁၄ ရက် မှာ ဘန်ကောက် ကို သွား မယ် ့US123 မှာ ပါ ။ ဟုတ် လား ။
ကား မှ ကား ဘီး ကို ဖြုတ် လိုက် သည် ။
ကျွန်တော် သိ ပါ ရ စေ ။
ဘူတာရုံ က အလွန်တရာ ပြည့် ကျပ် နေ သည် ။
```

[28]:
```
!wc ./output/otest.myword
```

```
  1000  14471 181791 ./output/otest.myword
```

0.8 Word Segmentation with CRF

[30]:
```
!time python ./baseline_segmenter.py --input ./data/otest.1k.word.input␣
↪--output ./output/otest.crf -m crf
```

```
2025-07-26 03:18:42.946682: I tensorflow/core/util/port.cc:110] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-07-26 03:18:42.947627: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 03:18:42.968129: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 03:18:42.968376: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
2025-07-26 03:18:43.350557: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT

real    0m1.360s
user    0m1.903s
sys     0m2.000s
```

[31]:
```
!head ./output/otest.crf
```

```
တစ် ကိုက် ကို ဝမ်ခု နှစ်ထောင် ပါ ။
မ နှစ် က သူ ကျွန်မ ကို သင်ပေး တယ် ။
ကျွန်တော့် ခုံ သွား ရှာ မ လို့ ။
အ တန်း စတာ ကြာ ပြီ လား ။
ဆေးနည်းနည်းစား လိုက် ၊ သုံး လေးရက်လောက် အနားယူ လိုက် ရင် ပျောက် သွား မှာ ပါ ။
အေးချမ်း မှု နဲ့ စည်းကမ်း ကို တည်မြဲ အောင် ထိန်း သိမ်း သည် ။
ဇွန်း ကို လို အပ် တယ် ။
ဘွဲ့ ရ ရင် ဘာ လုပ် မ လို့ လဲ ။
ကျွန်တော်ချောင်း ဆိုးခြင်း အတွက် တစ် ခု ခု လို ချင် တယ် ။
အ သီး အ နှံ တို့ မှ လွဲလျှင် လူ တို့ ၏ အဓိက အ စား အစာ မှာ ငါး ဖြစ် သည် ။
```

[36]:
```
!tail ./output/otest.crf
```

```
အိုးခွက်ပန်းကန် တွေ သိပ် မ ရှိ လို့ ထမင်း ဟင်းချက် ရ တာ အဆင်မ ပြေ ဘူး ။
စိတ်ဝင်စား ဖို့ ကောင်း တယ် ။
```

ဒီ ဆေး ကို တဝက်စီ ခွဲ ပေး ပါ ။
ရောင်းကောင်း လား ။
ဆရာဒီ သွား က ခဏခဏ နာ နေ တယ် ။
အခု ဘာ လုပ် နေ လဲ ။
ဇူလိုင် ၁၄ ရက် မှာ ဘန်ကောက် ကို သွား မယ့်US123 မှာ ပါ ။ဟုတ် လား ။
ကား မှ ကားဘီး ကို ဖြုတ် လိုက် သည် ။
ကျွန်တော် သိ ပါရစေ ။
ဘူတာ ရုံ က အ လွန်တရာ ပြည့် ကျပ် နေ သည် ။

[32]:
```
!wc ./output/otest.crf
```

```
  1000  13487 180807 ./output/otest.crf
```

0.9 Word Segmentation with LSTM

[33]:
```
!time python ./baseline_segmenter.py --input ./data/otest.1k.word.input␣
↪--output ./output/otest.lstm -m lstm
```

```
2025-07-26 03:19:54.974776: I tensorflow/core/util/port.cc:110] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-07-26 03:19:54.975704: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 03:19:54.996012: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 03:19:54.996267: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
2025-07-26 03:19:55.378127: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT
2025-07-26 03:19:55.866804: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:995]
successful NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
2025-07-26 03:19:55.885154: W
tensorflow/core/common_runtime/gpu/gpu_device.cc:1960] Cannot dlopen some GPU
libraries. Please make sure the missing libraries mentioned above are installed
properly if you would like to use GPU. Follow the guide at
https://www.tensorflow.org/install/gpu for how to download and setup the
required libraries for your platform.
Skipping registering GPU devices…
```

```
real    0m25.262s
user    0m26.509s
sys     0m3.656s
```

[34]:
```
!head ./output/otest.lstm
```

```
တစ် ကိုက် ကို ဝမ် ခု နှစ် ထောင် ပါ ။
မနှစ် က သူ ကျွန်မ ကို သင်ပေး 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 ။
```

[35]:
```
!tail ./output/otest.lstm
```

```
အိုးခွက် ပန်းကန် တွေ သိပ် မ ရှိ လို့ ထမင်း ဟင်း 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 ။
ဇူလိုင် ၁၄ ရက် မှာ ဘန်ကောက် ကို သွား မယ့် US 123 မှာ ပါ ။ ဟုတ် လား ။
ကား မှ ကား ဘီး ကို ဖြုတ် လိုက် သည် ။
ကျွန်တော် သိ ပါရစေ ။
ဘူတာရုံ က အလွန်တရာ ပြည့်ကျပ် နေ သည် ။
```

0.10 Evaluation for myWord

ဒီနေရာမှာ evaluate.py နဲ့ eval_segmentation.py ပရိဂရမ်နှစ်မျိုးလုံးသုံးပြီး evaluation လုပ်ပါမယ်။
evaluate.py ကတော့ 2006 ကတည်းက Yue Zhang (Computing lab, University of Oxford.) က ရေးခဲ့ပြီး word segmentation evaluation အတွက်သုံးခဲ့တဲ့ ပရိဂရမ်ပါ။
သူကို run တဲ့အခါမှာ hyp ဖိုင်ကို အရင် ပေးရပါတယ်။ ပြီးတော့ python2.7 နဲ့ run ပါ။

myWord အတွက် အရင်ဆုံး evaluation လုပ်ပါမယ်။

[42]:
```
!python2.7 ./evaluate.py ./output/otest.myword ./data/otest.1k.word
```

```
Tag precision: 0.856264252643
```

[43]:
```
!python ./eval_segmentation.py -r ./data/otest.1k.word -H ./output/otest.myword␣
↪--top-k 10
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                0.8489
Word Recall                   0.9121
Word F1-score                 0.8793
------------------------------------------------------------
Boundary Precision            0.4769
Boundary Recall               0.5124
Boundary F1-score             0.4940
------------------------------------------------------------
Vocab Precision               0.8911
Vocab Recall                  0.7645
Vocab F1-score                0.8230
============================================================

Additional Statistics:
Reference words: 13468
Hypothesis words: 14471
Correct words: 12284
Reference vocabulary size: 2709
Hypothesis vocabulary size: 2324
Common vocabulary: 2071

Top Segmentation Errors Analysis
============================================================
Total errors: 6568

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:
   55 × REF: '။' → HYP: 'သည်'
   53 × REF: '။' → HYP: 'တယ်'
   37 × REF: '။' → HYP: 'လား'
   33 × REF: '။' → HYP: 'ပါ'
   27 × REF: '။' → HYP: 'လဲ'
   22 × REF: 'မလား' → HYP: 'မ'
   22 × REF: '။' → HYP: 'မ'
   21 × REF: 'တယ်' → HYP: 'ပါ'
   20 × REF: 'ခုနှစ်' → HYP: 'ခု'
```

14 × REF: 'ဖလဲ' → HYP: 'ဖ'

0.11 Evaluation for CRF

[44]:
```
!python2.7 ./evaluate.py ./output/otest.crf ./data/otest.1k.word
```

```
Tag precision: 0.72239934752
```

[45]:
```
!python ./eval_segmentation.py -r ./data/otest.1k.word -H ./output/otest.crf␣
↪--top-k 10
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                 0.7019
Word Recall                    0.7029
Word F1-score                  0.7024
------------------------------------------------------------
Boundary Precision             0.3210
Boundary Recall                0.3215
Boundary F1-score              0.3213
------------------------------------------------------------
Vocab Precision                0.4646
Vocab Recall                   0.5482
Vocab F1-score                 0.5030
============================================================

Additional Statistics:
Reference words: 13468
Hypothesis words: 13487
Correct words: 9467
Reference vocabulary size: 2709
Hypothesis vocabulary size: 3196
Common vocabulary: 1485

Top Segmentation Errors Analysis
============================================================
Total errors: 8545

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:
   40 × REF: 'ပါ' → HYP: '။'
   37 × REF: '။' → HYP: 'တယ်'
```

```
36 × REF: 'တယ်' → HYP: '။'
32 × REF: 'သည်' → HYP: '။'
30 × REF: '။' → HYP: 'သည်'
21 × REF: 'လဲ' → HYP: '။'
15 × REF: '။' → HYP: 'ပါ'
13 × REF: 'လား' → HYP: '။'
10 × REF: '။' → HYP: 'လား'
10 × REF: '။' → HYP: 'လဲ'
```

0.12 Evaluation for LSTM

LSTM approach မော်ဒယ်နဲ့ ဖြတ်ပြီးရလာတဲ့ မြန်မာစာ စာကြောင်းတွေကိုလည်း evaluation လုပ်ကြည့်ရအောင်။

[46]:
```
!python2.7 ./evaluate.py ./output/otest.lstm ./data/otest.1k.word
```

```
Tag precision: 0.906617326948
```

[47]:
```
!python ./eval_segmentation.py -r ./data/otest.1k.word -H ./output/otest.lstm␣
↪--top-k 10
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                 0.9055
Word Recall                    0.9266
Word F1-score                  0.9159
------------------------------------------------------------
Boundary Precision             0.6025
Boundary Recall                0.6165
Boundary F1-score              0.6094
------------------------------------------------------------
Vocab Precision                0.8091
Vocab Recall                   0.8169
Vocab F1-score                 0.8130
============================================================

Additional Statistics:
Reference words: 13468
Hypothesis words: 13782
Correct words: 12479
Reference vocabulary size: 2709
Hypothesis vocabulary size: 2735
Common vocabulary: 2213

Top Segmentation Errors Analysis
```

```
============================================================
Total errors: 5042

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:
   46 × REF: '။' → HYP: 'တယ်'
   38 × REF: '။' → HYP: 'သည်'
   21 × REF: 'သည်' → HYP: '။'
   14 × REF: 'တယ်' → HYP: '။'
   13 × REF: '။' → HYP: 'မယ်'
   13 × REF: 'တယ်' → HYP: 'ပါ'
   13 × REF: 'ပါ' → HYP: '။'
   12 × REF: '။' → HYP: 'လား'
   12 × REF: 'သည်' → HYP: 'ဖြစ်'
   12 × REF: '။' → HYP: 'ပါ'
```

0.13 Preparing a Big Test Data

ဒီတစ်ခါတော့ word segmenter တွေရဲ့ speed ကိုပါ တိုင်းတာချင်တဲ့အတွက် စာကြောင်းရေ 10K (တစ်သောင်း) ရှိတဲ့ test data ကို ဖန်တီးမယ်။
ပြီးတော့ မြန်မာ punctuation တွေလည်း ဖြုတ်ထားခဲ့ပြီး စမ်းမယ်။

[49]:
```
!pwd
```

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

[50]:
```
!wget https://raw.githubusercontent.com/ye-kyaw-thu/myPOS/refs/heads/master/
 ↪corpus-ver-3.0/corpus/mypos-ver.3.0.shuf.notag.nopunc.txt
```

```
--2025-07-26 16:46:28--  https://raw.githubusercontent.com/ye-kyaw-
thu/myPOS/refs/heads/master/corpus-ver-3.0/corpus/mypos-
ver.3.0.shuf.notag.nopunc.txt
Resolving raw.githubusercontent.com (raw.githubusercontent.com)…
185.199.109.133, 185.199.111.133, 185.199.110.133, …
Connecting to raw.githubusercontent.com
(raw.githubusercontent.com)|185.199.109.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 7303817 (7.0M) [application/octet-stream]
Saving to: ‘mypos-ver.3.0.shuf.notag.nopunc.txt’

mypos-ver.3.0.shuf. 100%[==================>]   6.96M  25.9MB/s    in 0.3s

2025-07-26 16:46:29 (25.9 MB/s) - ‘mypos-ver.3.0.shuf.notag.nopunc.txt’ saved
[7303817/7303817]
```

Download လုပ်ယူထားတဲ့ ဖိုင်ထဲက ထိပ်ဆုံးပိုင်းစာကြောင်း တစ်သောင်းကို ဆွဲထုတ်ယူပြီး **10k_test.txt** ဖိုင်အဖြစ် သိမ်းခဲ့တယ်။

[51]:
```
!head -n 10000 ./mypos-ver.3.0.shuf.notag.nopunc.txt > 10k_test.txt
```

[52]:
```
!mv ./10k_test.txt ./data/
```

ဒီဖိုင်မှာက POS tag တွေက ဖြုတ်ထားပြီးသားပါ။ အဲဒါကြောင့် POS tag ဖြုတ်ရတဲ့ အလုပ်တော့ လုပ်ဖို့ မလိုအပ်ပါဘူး။

[53]:
```
!head ./data/10k_test.txt
```

၁၉၆၂ ခုနှစ် ခန့်မှန်း သန်းခေါင်စာရင်း အရ လူဦးရေ ၁၁၅၃၃၁ ယောက် ရှိ သည်
လူ တိုင်း တွင် သင့်မြတ် လျော်ကန် စွာ ကန့်သတ် ထား သည့် အလုပ် လုပ် ချိန် 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 ပိဋ္ဌကတ်သုံးပုံ စာပေ တွေ ကို စာစစ် သံဃာတော်ကြီး တွေ ရဲ့ ရှေ့ မှာ အလွတ် ပြန် ပြီး ရွတ်ပြ ရ တာ ပေါ့
ဒီ မှာ ကျွန်တော့် သက်သေခံကတ် ပါ
၂၀ ရာစု မြန်မာ့ သမိုင်း သန်းဝင်းလှိုင် ၂၀၀၉ ခု မေ လ ကံကော်ဝတ်ရည် စာပေ
ကျွန်တော် မျက်မှန် တစ် လက် လုပ် 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

0.14 Preprocessing

myTokenize ကို input လုပ်တဲ့အခါမှာ space တွေအကုန် ဖြုတ်ပေးထားရတာမို့ smart_space_remover.py ကို သုံးမယ်။

[55]:
```
cd data
```

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

[57]:
```
!python ./smart_space_remover.py --input ./10k_test.txt --output ./10k_test.
↪input --mode my_not_num
```

Space ဖြုတ်ပြီး word segmenter ကို pass လုပ်မယ့် test input ဖိုင်ကို ကြည့်ကြည့်ရအောင်။

[58]:
```
!head ./10k_test.input
```

၁၉၆၂ ခုနှစ်ခန့်မှန်းသန်းခေါင်စာရင်းအရလူဦးရေ ၁၁၅၃၃၁ ယောက်ရှိသည်
လူတိုင်းတွင်သင့်မြတ်လျော်ကန်စွာကန့်သတ်ထားသည့်အလုပ်လုပ်ချိန်အပြင်လစာနှင့်တကွအခါကာ

လအား လျော်စွာသတ်မှတ်ထားသည့်အလုပ်အား လပ်ရက်များပါဝင်သည့်အနား ယူခွင့်နှင့်အား လပ်ခွင့်
ခံစားပိုင်ခွင့်ရှိသည်
ဤနည်းကိုစစ်ယူသောနည်းဟုခေါ်သည်
စာပြန်ပွဲဆိုတာကအာဂုံဆောင်အလွတ်ကျက်ထားတဲ့ပိဋကတ်သုံးပုံစာပေတွေကိုစာစစ်သံဃာတော်ကြီး
တွေရဲ့ရှေ့မှာအလွတ်ပြန်ပြီးရွတ်ပြရတာပေါ့
ဒီမှာကျွန်တော့်သက်သေခံကတ်ပါ
၂၀ရာစုမြန်မာ့သမိုင်းသန်းဝင်းလှိုင် ၂၀၀၉ ခုမေလကံကော်ဝတ်ရည်စာပေ
ကျွန်တော်မျက်မှန်တစ်လက်လုပ်ချင်ပါတယ်
ကျွန်တော်တို့ကဒီအမှုရဲ့ကြံရာပါကိုဖမ်းမိဖို့ကြိုးစားခဲ့တယ်
ကလေးမီးဖွားဖို့ခန့်မှန်းရက်ကဘယ်တော့ပါလဲ
အရိုးရှင်းဆုံးကာဗိုဟိုက်ဒရိတ်မှာဂလူးကို့စ်ဂလက်တို့စ်ဖရုပ်တို့စ်စသည့်မိုနိုဆက်ကရို
က်များဖြစ်သည်

0.15 Word Segmentation with myWord

[59]:
```
cd ..
```

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

[63]:
```
!time python ./baseline_segmenter.py --input ./data/10k_test.input --output ./
↪output/10k_test.myword -m myword
```

```
2025-07-26 17:25:57.185836: I tensorflow/core/util/port.cc:110] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-07-26 17:25:57.186801: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 17:25:57.207083: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 17:25:57.207334: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
2025-07-26 17:25:57.607601: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT

real    89m50.495s
user    81m56.812s
sys     7m55.380s
```

[64]:
```
!wc ./output/10k_test.myword
```

```
  10000  128206 1696728 ./output/10k_test.myword
```

[65]:
```
!head ./output/10k_test.myword
```

၁၉၆၂ ခု နှစ် ခန့်မှန်း သန်းခေါင်စာရင်း အရ လူ ဦး ရေ ၁၁၅၃၁၁ ယောက် ရှိ သည်
လူ တိုင်း တွင် သင့်မြတ် လျော်ကန် စွာ ကန့်သတ် ထား သည့် အလုပ် လုပ် ချိန် အပြင် လစာ
နှင့်တကွ အခါ ကာလ အားလျော်စွာ သတ်မှတ် ထား သည့် အလုပ် အားလပ်ရက် များ ပါ ဝင်
သည့်
အနားယူ ခွင့် နှင့် အားလပ် ခွင့် ခံစားပိုင်ခွင့် ရှိ သည်
ဤ နည်း ကို စစ် ယူ သော နည်း ဟု ခေါ် သည်
စာ ပြန် ပွဲ ဆို တာ က အာဂုံဆောင် အလွတ်ကျက် ထား တဲ့ ပိဋကတ်သုံးပုံ စာပေ တွေ ကို စာ
စစ် သံဃာတော် ကြီး တွေ ရဲ့ ရှေ့ မှာ အလွတ် ပြန် ပြီး ရွတ် ပြ ရ တာ ပေါ့
ဒီ မှာ ကျွန်တော့် သက်သေခံ ကတ် ပါ
၂၀ ရာ စု မြန်မာ့ သမိုင်း သန်းဝင်းလှိုင် ၂၀၀၉ ခု မေ လ ကံကော်ဝတ်ရည် စာပေ
ကျွန်တော် မျက်မှန် တစ် လက် လုပ် ချင် ပါ တယ်
ကျွန်တော် တို့ က ဒီ အမှု ရဲ့ ကြံ ရာ ပါ ကို ဖမ်း မိ ဖို့ ကြိုးစား ခဲ့ တယ်
က လေး မီးဖွား ဖို့ ခန့်မှန်း ရက် က ဘယ် တော့ ပါ လဲ
အရိုးရှင်းဆုံး ကာဗိုဟိုက်ဒရိတ် မှာ ဂလူးကို့စ် ဂလက်တို့စ် ဖရုပ်တို့စ် စ သည့်
မိုနိုဆက်ကရိုက် များ ဖြစ် သည်

[66]:
```
!tail ./output/10k_test.myword
```

ကျွန်တော် စိန် နဲ့ တစ် ခု လို ချင် ပါ တယ်
ကြိုး ကြာ
သူ အဘိဓာန်စာလုံး ရှာ တတ် ပါ တယ်
ဩော် ကျွန်တော် မျက်စိလည် နေ ပြီ ထင် တယ် ဒီ နေ ရာ က နေ ဘူတာရုံ ကို ဘယ်လို သွား
ရ မ
လဲ
လော့အိန်ဂျလိစ် က နေ တိုကျို ကို လေယာဉ် ခ ဘယ်လောက် လဲ
ပထမဆုံး အဆင်ပြေ မယ့် လေယာဉ် လို ချင် ပါ တယ်
ရေ ထွက် ပစ္စည်း များ ထုတ် လုပ် မှု တွင် လည်း တန် ချိန် ၄၁.၂၂၄ သန်း ထိ ၁၉၇၈ ခု
နှစ် ထက် ၈.၈ ဆ တိုးတက် ထုတ် လုပ် နိုင် ခဲ့ သည်
ဟုတ် တယ် ဟင်းသီးဟင်းရွက် သွား ဝယ် ရ အောင်
ဆရာ ကြီး က ၄ နာရီ လောက် ဆို အလုပ် နည်းနည်း ရှင်း ပြီ ၅ နာရီ မှ ရုံး ဆင်း မှာ ဆို
တော့ တစ် နာရီ လောက် တော့ တွေ့ ချိန် ရ မှာ ပဲ
ဝင် ခွင့် ဈေးနှုန်း ရော ပါ လား

0.16 Word Segmentation with CRF

[67]:
```
!time python ./baseline_segmenter.py --input ./data/10k_test.input --output ./
output/10k_test.crf -m crf
```

```
2025-07-26 19:40:39.289427: I tensorflow/core/util/port.cc:110] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
```

```
2025-07-26 19:40:39.290426: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 19:40:39.310621: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 19:40:39.310889: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
2025-07-26 19:40:39.692507: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT

real    0m2.641s
user    0m3.190s
sys     0m1.993s
```

[68]:
```
!head ./output/10k_test.crf
```

```
၁၉၆၂ ခုနှစ် ခန့်မှန်းသန်းခေါင်စာရင်း အ ရ လူဦးရေ ၁၁၅၃၃၁ ယောက်ရှိ သည်
လူ တိုင်း တွင် သင့်မြတ်လျော်ကန် စွာ ကန့်သတ်ထား သည့် အလုပ်လုပ်ချိန် အပြင်လစာ
နှင့်တကွ အ ခါ ကာလ အားလျော် စွာ သတ် မှတ်ထား သည့် အလုပ် အားလပ် ရက် များ ပါဝင်
↪သည့်
အနားယူ ခွင့် နှင့် အားလပ်ခွင့် ခံစား ပိုင်ခွင့် ရှိ သည်
ဤနည်း ကို စစ်ယူ သော နည်း ဟု ခေါ် သည်
စာပြန်ပွဲ ဆို တာ က အာဂုံ ဆောင် အ လွတ်ကျက်ထား တဲ့ ပိဋ္ဌကတ်သုံးပုံစာ ပေ တွေ ကို
စာစစ်သံဃာတော်ကြီး တွေ ရဲ့ ရှေ့ မှာ အ လွတ်ပြန် ပြီး ရွတ်ပြ ရ တာ ပေါ့
ဒီ မှာ ကျွန်တော့် သက် သေခံ ကတ် ပါ
၂၀ရာစု မြန်မာ့ သမိုင်းသန်းဝင်း လှိုင် ၂၀၀၉ ခုမေ လ ကံကော်ဝတ်ရည်စာပေ
ကျွန်တော် မျက် မှန် တစ် လက်လုပ်ချင် ပါ တယ်
ကျွန်တော် တို့ က ဒီ အမှု ရဲ့ ကြံ ရာ ပါ ကို ဖမ်း မိ ဖို့ကြိုးစား ခဲ့ တယ်
ကလေး မီးဖွား ဖို့ခန့် မှန်းရက် က ဘယ်တော့ ပါ လဲ
အရိုးရှင်းဆုံး ကာ ဗိုဟိုက်ဒရိတ်မှာ ဂ လူး ကို့စ်ဂလက်တို့စ်ဖ ရပ်တို့စ် စသည့် မို
နို ဆက်ကရိုက် များ ဖြစ် သည်
```

[69]:
```
!tail ./output/10k_test.crf
```

```
ကျွန်တော်စိန် နဲ့ တစ် ခု လို ချင် ပါ တယ်
ကြိုးကြာ
သူ အ ဘိဓာန်စာလုံး ရှာ တတ် ပါ တယ်
သြော် ကျွန်တော် မျက်စိ လည် နေ ပြီ ထင် တယ် ဒီ နေရာ က နေ ဘူတာ ရုံ ကို ဘယ်လို ပ
↪သွား ရ
မလဲ
လော့ အိန်ဂျလိစ် က နေ တိုကျိုကို လေယာဉ်ခ ဘယ်လောက် လဲ
ပထမဆုံး အ ဆင်ပြေ မယ့် လေယာဉ် လို ချင် ပါ တယ်
```

ရေထွက် ပစ္စည်း များ ထုတ်လုပ် မှု တွင် လည်းတန်ချိန် ၄၁.၂၄ သန်းထိ ၁၉၇၈ ခုနှစ်
ထက်၈.၈ ဆ တိုးတက်ထုတ်လုပ် နိုင် ခဲ့ သည်
ဟုတ် တယ် ဟင်းသီးဟင်းရွက် သွား ဝယ် ရ အောင်
ဆရာကြီး ကဲ့ နာရီ လောက်ဆို အလုပ် နည်းနည်း ရှင်း ပြီ ၅ နာရီ မှ ရုံးဆင်း မှာ ဆို
တော့ တစ် နာရီ လောက်တော့ တွေ့ချိန် ရ မှာ ပဲ
ဝင်ခွင့် ဈေးနှုန်း ရော ပါ လား

0.17 Word Segmentation with LSTM

စာကြောင်းရေ တစ်သောင်း ရှိတဲ့ test ဖိုင်ကို ဒီတခါတော့ LSTM မော်ဒယ်နဲ့ word segmentation လုပ်မယ်။

[70]:
```
!time python ./baseline_segmenter.py --input ./data/10k_test.input --output ./
↪output/10k_test.lstm -m lstm
```

```
2025-07-26 19:42:09.683039: I tensorflow/core/util/port.cc:110] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-07-26 19:42:09.683983: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 19:42:09.704895: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not
find cuda drivers on your machine, GPU will not be used.
2025-07-26 19:42:09.705143: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 AVX_VNNI FMA, in other operations,
rebuild TensorFlow with the appropriate compiler flags.
2025-07-26 19:42:10.087859: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not
find TensorRT
2025-07-26 19:42:10.583264: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:995]
successful NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
2025-07-26 19:42:10.604644: W
tensorflow/core/common_runtime/gpu/gpu_device.cc:1960] Cannot dlopen some GPU
libraries. Please make sure the missing libraries mentioned above are installed
properly if you would like to use GPU. Follow the guide at
https://www.tensorflow.org/install/gpu for how to download and setup the
required libraries for your platform.
Skipping registering GPU devices…

real    3m39.747s
user    3m46.339s
sys     0m17.206s
```

[71]:
```
!head ./output/10k_test.lstm
```

၁၉၆၂ ခုနှစ် ခန့်မှန်း သန်းခေါင်စာရင်း အရ လူဦးရေ ၁၁၅၃၃၁ ယောက် ရှိ သည်
လူ တိုင်း တွင် သင့်မြတ် လျော်ကန် စွာ ကန့်သတ် ထား သည့် အလုပ် လုပ် ချိန် အပြင် လစာ
နှင့်တကွ အခါ ကာလ အားလျော် စွာ သတ်မှတ် ထား သည့် အလုပ်အားလပ်ရက် များ ပါဝင် သည့်
အနားယူ ခွင့် နှင့် အားလပ်ခွင့် ခံစား ပိုင် ခွင့် ရှိ သည်
ဤ နည်း ကို စစ်ယူ သော နည်း ဟု ခေါ် သည်
စာပြန် ပွဲ ဆို တာ က အာဂုံ ဆောင် အလွတ်ကျက် ထား တဲ့ ပိဋ္ဌကတ် သုံး ပုံ စာပေ တွေ ကို
စာစစ် သံဃာတော် ကြီး တွေ ရဲ့ ရှေ့ မှာ အလွတ် ပြန် ပြီး ရွတ်ပြ ရ တာ ပေါ့
ဒီ မှာ ကျွန်တော့် သက် သေ ခံကတ် ပါ
၂၀ ရာစု မြန်မာ့ သမိုင်း သန်းဝင်းလှိုင် ၂၀၀ ၉ ခုမေ လ ကဲ ကော်ဝတ်ရည် စာပေ
ကျွန်တော် မျက်မှန် တစ် လက် လုပ် ချင် ပါ တယ်
ကျွန်တော် တို့ က ဒီ အမှု ရဲ့ ကြဲ ရာ ပါ ကို ဖမ်း မိ ဖို့ ကြိုးစား ခဲ့ တယ်
ကလေး မီးဖွား ဖို့ ခန့်မှန်း ရက် က ဘယ်တော့ ပါ လဲ
အရိုးရှင်းဆုံး ကာဗိုဟိုက်ဒရိတ် မှာ ဂလူးကို့စ် ဂလက်တို့စ်ဖရုပ် တို့စ် စသည့်
မိုနိုဆက် က ရိုက် များ ဖြစ် သည်

[72]:
```
!tail ./output/10k_test.lstm
```

ကျွန်တော်စိန် နဲ့ တစ် ခု လို ချင် ပါ တယ်
ကြိုး ကြာ
သူ အဘိဓာန် စာလုံး ရှာ တတ် ပါ တယ်
သြော် ကျွန်တော် မျက်စိ လည် နေ ပြီ ထင် တယ် ဒီ နေရာ က နေ ဘူတာရုံ ကို ဘယ်လို သွား␣
↪ရ မလဲ
လော့အိန်ဂျလိစ် က နေ တိုကျို ကို လေယာဉ် ခ ဘယ်လောက် လဲ
ပထမဆုံး အဆင်ပြေ မယ့် လေယာဉ် လို ချင် ပါ တယ်
ရေထွက် ပစ္စည်း များ ထုတ်လုပ် မှု တွင် လည်း တန်ချိန် ၄၁.၂၄ သန်း ထိ ၁၉၇၈ ခုနှစ်
ထက် ၈.၈ ဆ တိုးတက် ထုတ်လုပ် နိုင် ခဲ့ သည်
ဟုတ် တယ် ဟင်းသီးဟင်းရွက် သွား ဝယ် ရ အောင်
ဆရာကြီး က ၄ နာရီ လောက် ဆို အလုပ် နည်းနည်း ရှင်း ပြီ ၅ နာရီ မှ ရုံး ဆင်း မှာ ဆို
တော့ တစ် နာရီ လောက် တော့ တွေ့ ချိန် ရ မှာ ပဲ
ဝင်ခွင့် ဈေးနှုန်း ရော ပါ လား

0.18 Evaluation for Big Test Dataset

ဒီ ဒေတာ က တကယ်ကတော့ closed-test လို့ ယူဆလို့ ရပါတယ်။
ထပ်မှာမယ်။ evaluate.py နဲ့ evaluation လုပ်တဲ့အခါမှာ hypothesis ဖိုင်ကို အရင်ပေးပါ။ ပြီးမှ reference (manually segmented file) ကို argument အနေနဲ့ ပေးပြီး run ပါ။

အရင်ဆုံး myWord နဲ့ ဖြတ်ပြီးထွက်လာတဲ့ output ကို evaluation လုပ်ပါမယ်။

[75]:
```
!python2.7 ./evaluate.py ./output/10k_test.myword ./data/10k_test.txt
```

```
Tag precision: 0.834305726721
```

ဆရာရေးထားတဲ့ eval_segmentation.py ကို run တဲ့အခါမှာတော့ -r, -H နဲ့ အစီအစဉ်က သတ်မှတ်လို့ ပါတယ်။

[76]:
```
!python ./eval_segmentation.py -H ./output/10k_test.myword -r ./data/10k_test.
↪txt --top-k 30
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                 0.8235
Word Recall                    0.8958
Word F1-score                  0.8581
------------------------------------------------------------
Boundary Precision             0.4695
Boundary Recall                0.5107
Boundary F1-score              0.4893
------------------------------------------------------------
Vocab Precision                0.8953
Vocab Recall                   0.6456
Vocab F1-score                 0.7502
============================================================

Additional Statistics:
Reference words: 117857
Hypothesis words: 128206
Correct words: 105579
Reference vocabulary size: 10840
Hypothesis vocabulary size: 7816
Common vocabulary: 6998

Top Segmentation Errors Analysis
============================================================
Total errors: 57570

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:
  176 × REF: 'မလား' → HYP: 'မ'
  172 × REF: 'မလဲ' → HYP: 'မ'
  170 × REF: 'တယ်' → HYP: 'ပါ'
  148 × REF: 'နေရာ' → HYP: 'နေ'
  128 × REF: 'ခုနှစ်' → HYP: 'ခု'
   97 × REF: 'သည်' → HYP: 'ခဲ့'
   95 × REF: 'သည်' → HYP: 'ဖြစ်'
```

```
88 × REF: 'ရထား' → HYP: 'ရ'
73 × REF: 'တယ်' → HYP: 'ခဲ့'
71 × REF: 'ကလေး' → HYP: 'က'
67 × REF: 'ကို' → HYP: 'များ'
62 × REF: 'ဘာသာ' → HYP: 'ဘာ'
62 × REF: 'သည်' → HYP: 'ရှိ'
62 × REF: 'သည်' → HYP: 'ကြ'
59 × REF: 'တယ်' → HYP: 'နေ'
59 × REF: 'သူမ' → HYP: 'သူ'
58 × REF: 'သည်' → HYP: 'များ'
58 × REF: 'ရ' → HYP: 'လို့'
57 × REF: 'ပါ' → HYP: 'ပေး'
55 × REF: 'မှာ' → HYP: 'ရာ'
55 × REF: 'ခု' → HYP: 'တစ်'
54 × REF: 'သည်' → HYP: 'ပါ'
53 × REF: 'ပါ' → HYP: 'မ'
50 × REF: 'ဘယ်သူ' → HYP: 'ဘယ်'
49 × REF: 'ပါ' → HYP: 'ရှိ'
48 × REF: 'တယ်' → HYP: 'ရှိ'
48 × REF: 'ဘူး' → HYP: 'မ'
47 × REF: 'ပါ' → HYP: 'ချင်'
47 × REF: 'သည်' → HYP: 'ရ'
44 × REF: 'တွင်' → HYP: 'နှစ်'
```

CRF နဲ့ ဖြတ်ထားတဲ့ ဖိုင်ကို evaluation လုပ်ကြည့်မယ်။

[77]:
```
!python2.7 ./evaluate.py ./output/10k_test.crf ./data/10k_test.txt
```

```
Tag precision: 0.704589402575
```

[78]:
```
!python ./eval_segmentation.py -H ./output/10k_test.crf -r ./data/10k_test.txt␣
↪--top-k 30
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                 0.6928
Word Recall                    0.6961
Word F1-score                  0.6944
------------------------------------------------------------
Boundary Precision             0.3186
Boundary Recall                0.3201
Boundary F1-score              0.3194
------------------------------------------------------------
```

```
Vocab Precision               0.3256
Vocab Recall                  0.4674
Vocab F1-score                0.3838
============================================================

Additional Statistics:
Reference words: 117857
Hypothesis words: 118425
Correct words: 82042
Reference vocabulary size: 10840
Hypothesis vocabulary size: 15563
Common vocabulary: 5067

Top Segmentation Errors Analysis
============================================================
Total errors: 74443

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:
  152 × REF: 'ပါ' → HYP: 'တယ်'
  114 × REF: 'တယ်' → HYP: 'ပါ'
   74 × REF: 'အတွက်' → HYP: 'အ'
   70 × REF: 'သည်' → HYP: 'ဖြစ်'
   67 × REF: 'ဘယ်' → HYP: 'ဘယ်မှာ'
   67 × REF: 'သည်' → HYP: 'ခဲ့'
   63 × REF: 'လို့' → HYP: 'ရ'
   60 × REF: 'ဘယ်' → HYP: 'ဘယ်အချိန်'
   58 × REF: 'ဖြစ်' → HYP: 'သည်'
   56 × REF: 'မှာ' → HYP: 'လဲ'
   56 × REF: 'ကို' → HYP: 'အ'
   56 × REF: 'ကို' → HYP: 'များ'
   54 × REF: 'ခု' → HYP: 'တစ်'
   54 × REF: 'တယ်' → HYP: 'ခဲ့'
   53 × REF: 'နိုင်' → HYP: 'နိုင်မလား'
   52 × REF: 'သည်' → HYP: 'များ'
   50 × REF: 'ပါ' → HYP: 'ပေး'
   49 × REF: 'ခဲ့' → HYP: 'သည်'
   49 × REF: 'ချင်' → HYP: 'တယ်'
   48 × REF: 'ကိစ္စ' → HYP: 'ကိစ္'
   47 × REF: 'ရှိ' → HYP: 'ပါ'
   47 × REF: 'များ' → HYP: 'ကို'
```

```
46 × REF: 'ရှိ' → HYP: 'မ'
44 × REF: 'ပေး' → HYP: 'ပါ'
43 × REF: 'ချင်' → HYP: 'ပါ'
43 × REF: 'ရ' → HYP: 'လို့'
41 × REF: 'ပါ' → HYP: 'ရှိ'
40 × REF: 'မနက်ဖြန်' → HYP: 'မနက်'
39 × REF: 'နေ' → HYP: 'တယ်'
39 × REF: 'ကြ' → HYP: 'သည်'
```

Neural network approach ဖြစ်တဲ့ lstm နဲ့ ဖြတ်ထားတဲ့ output ဖိုင်ကိုလည်း evaluation လုပ်မယ်။

[79]:
```
!python2.7 ./evaluate.py ./output/10k_test.lstm ./data/10k_test.txt
```

```
Tag precision: 0.898953005049
```

[80]:
```
!python ./eval_segmentation.py -H ./output/10k_test.lstm -r ./data/10k_test.txt␣
↪--top-k 30
```

```
Word Segmentation Evaluation Results
============================================================
Metric                          Score
------------------------------------------------------------
Word Precision                 0.8970
Word Recall                    0.9210
Word F1-score                  0.9088
------------------------------------------------------------
Boundary Precision             0.6119
Boundary Recall                0.6283
Boundary F1-score              0.6200
------------------------------------------------------------
Vocab Precision                0.7265
Vocab Recall                   0.7093
Vocab F1-score                 0.7178
============================================================

Additional Statistics:
Reference words: 117857
Hypothesis words: 121013
Correct words: 108546
Reference vocabulary size: 10840
Hypothesis vocabulary size: 10584
Common vocabulary: 7689

Top Segmentation Errors Analysis
============================================================
Total errors: 42500
```

```
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:
  133 × REF: 'တယ်' → HYP: 'ပါ'
   91 × REF: 'သည်' → HYP: 'ဖြစ်'
   73 × REF: 'တယ်' → HYP: 'ခဲ့'
   73 × REF: 'သည်' → HYP: 'ခဲ့'
   57 × REF: 'ဖြစ်' → HYP: 'သည်'
   56 × REF: 'ပါ' → HYP: 'တယ်'
   53 × REF: 'ပါ' → HYP: 'ပေး'
   51 × REF: 'တယ်' → HYP: 'နေ'
   47 × REF: 'များ' → HYP: 'ကို'
   43 × REF: 'သည်' → HYP: 'ကြ'
   41 × REF: 'ခဲ့' → HYP: 'သည်'
   40 × REF: 'ပါ' → HYP: 'ရှိ'
   39 × REF: 'ခု' → HYP: 'တစ်'
   38 × REF: 'ကြ' → HYP: 'သည်'
   38 × REF: 'တယ်' → HYP: 'ရှိ'
   37 × REF: 'သည်' → HYP: 'ရှိ'
   36 × REF: 'ရ' → HYP: 'လို့'
   35 × REF: 'ရအောင်' → HYP: 'ရ'
   34 × REF: 'ကို' → HYP: 'များ'
   33 × REF: 'သည်' → HYP: 'များ'
   33 × REF: 'မလား' → HYP: 'နိုင်'
   32 × REF: 'သည်' → HYP: 'ရ'
   30 × REF: 'သွား' → HYP: 'ကို'
   30 × REF: 'သည်' → HYP: 'လေ'
   30 × REF: 'ရှိ' → HYP: 'မ'
   29 × REF: 'တယ်' → HYP: 'ရ'
   29 × REF: 'ပါ' → HYP: 'ချင်'
   29 × REF: 'သည်' → HYP: 'ပါ'
   27 × REF: 'ပါ' → HYP: 'ရ'
   26 × REF: 'ရှိ' → HYP: 'သည်'
```

0.19 Summary

လက်ရှိအချိန်ထိ Lab မှာ သုံးခဲ့ကြတဲ့ word segmenter သုံးမျိုးကို open-test, closed-test လုပ်ကြည့်ခဲ့တယ်။ တနည်းအားဖြင့် စာကြောင်း အရေအတွက် စုစုပေါင်း တစ်ထောင်ရှိတဲ့ open-test ဒေတာနဲ့ စာကြောင်း အရေအတွက် တစ်သောင်း ရှိတဲ့ closed-test ဒေတာတွေနဲ့ baseline experiment အနေနဲ့ လုပ်ကြည့်ခဲ့ပါတယ်။ ရည်ရွယ်ချက်ကတော့ အသစ် develop လုပ်ထားတဲ့ dag_bimm_segmenter ရဲ့ word segmentation performance, runtime တွေကို နှိုင်းယှဉ်ကြည့်ချင်လို့ပါ။

ရလဒ်တွေကတော့ အောက်ပါအတိုင်းပါ။

0.20 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**

0.21 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**

0.22 Vocab Metrics for 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

0.23 Additional Evaluation Statistics (Open-Test)

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

0.24 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**

0.25 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**

0.26 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

0.27 Additional Evaluation Statistics (Closed Test)

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

0.28 Summary

- LU Lab. ရဲ့ word segmentation model တွေဖြစ်တဲ့ myWord ရယ်၊ CRF ရယ်၊ LSTM သုံးမျိုးကို open-test ဒေတာ စာကြောင်းရေ တစ်ထောင်၊ closed-test ဒေတာ တစ်သောင်းနဲ့ formal evaluation လုပ်ကြည့်ခဲ့တယ်။
- သုံးမျိုးထဲမှာ ရလဒ်အကောင်းဆုံးကို ပေးနိုင်တာက LSTM ပါ
- ဝမ်းသာဖို့ကောင်းတာက myWord က အရမ်း strong ဖြစ်တဲ့ CRF ထက် ရလဒ်ပိုကောင်းတဲ့အချက်ကိုပါ
- ဒီ သုံးမျိုးကို baseline အနေနဲ့ ထားပြီး July 2025 မှာ အသစ်စမ်းထားတဲ့ oppaWord word segmenter နဲ့ နှိုင်းယှဉ်ကြည့်မယ်။

[ ]: