To run in Colab (skip otherwise)

```
!curl -sS https://apertium.projectjj.com/apt/install-release.sh | sudo
bash
!apt install apertium-all-dev lexd
```

Make sure that `hfst-guess bezhta.guesser.hfst` is in the same directory

Coverage

Unrecognized tokens

```
def guess_word(word):
    output = os.popen(f"echo {word} | hfst-guess bezhta.guesser.hfst -
n 100").read().rstrip('\n\n')
    parses = []
    for el in output.split('\n'):
        parses.append(':'.join(el.split('\t')))

    return parses

def check_guesser_coverage(path):
    with open(path, 'r', encoding='utf-8') as file:
        file = file.read()
        words = re.findall(r"(?P<num>\d+) \^(?P<word>[a-яӀāēīōūӏᴴ]
+)\/.*\$", file)
        wd = {}
        for word in words:
            guess = guess_word(word[1])
            if guess == ['']:
                guess = []
            wd[word[1]] = {'number': int(word[0]), 'guess':guess,
'len_guess':len(guess)}

    n_recog = 0
    n_unrecog = 0
    for word in wd.keys():
        if wd[word]['len_guess']:
            n_recog += wd[word]['number']
        else:
            n_unrecog += wd[word]['number']

    print('recog: ', n_recog)
    print('unrecog: ', n_unrecog)
    print('coverage: ', n_recog/(n_recog+n_unrecog))

    return wd
```

```
gospel = check_guesser_coverage('unrecog-gospel.txt')

recog:  4019
unrecog:  995
coverage:  0.8015556441962505

gospel = check_guesser_coverage('unrecog-gospel.txt')

recog:  4019
unrecog:  995
coverage:  0.8015556441962505

gospel = check_guesser_coverage('unrecog-gospel.txt')

recog:  4019
unrecog:  995
coverage:  0.8015556441962505

gospel = check_guesser_coverage('unrecog-gospel.txt')

recog:  4019
unrecog:  995
coverage:  0.8015556441962505
```

Full corpora

```
def check_guesser_coverage_full(path):
   with open(path, 'r', encoding='utf-8') as file:
    un = []
    total = 0.0
    tokens_recognised = 0.0
    unique = 0.0
    lines = file.readlines()
    for l in lines:
        parses = [guess_word(i.strip()) for i in l.split()]
        for w in parses:
          if w[0] != '':
            tokens_recognised += 1
          total +=1
    print('tokens: ', total)
    print('tokens recognised: ', tokens_recognised)
    print('tokens unrecognised: ', total - tokens_recognised)
    print('token coverage: ', tokens_recognised / total)

luke_full = check_guesser_coverage_full('text-Luke.txt')

tokens:  19739.0
tokens recognised:  16011.0
tokens unrecognised:  3728.0
token coverage:  0.8111353158721313
```

```
prov_full = check_guesser_coverage_full('text-Prov.txt')

tokens:  11644.0
tokens recognised:  9192.0
tokens unrecognised:  2452.0
token coverage:  0.7894194434902095

text_1_full = check_guesser_coverage_full('text-turkey.txt')

tokens:  86.0
tokens recognised:  71.0
tokens unrecognised:  15.0
token coverage:  0.8255813953488372

text_2_full = check_guesser_coverage_full('text-life.txt')

tokens:  116.0
tokens recognised:  90.0
tokens unrecognised:  26.0
token coverage:  0.7758620689655172
```

Accuracy

All tokens

```
def guess_word_acc(word):
    output = os.popen(f"echo {word} | hfst-guess bezhta.guesser.hfst -
n 100").read().rstrip('\n\n')
    parses = []
    for el in output.split('\n'):
      if output.split('\n')[0] != '':
        parse = el.split('\t')[1]
        parses.append(re.sub('\[GUESS_CATEGORY=\w+\]', '', parse))
    if parses == ['']:
        parses = []
    return parses

def check_guesser_acc(path):
  with open(path, 'r', encoding='utf-8') as file:
      file = file.read()
      fully_guessed = 0.0
      recog = 0.0
      total = 0.0
      tags_guessed = 0.0
      seen = []
      words = re.findall(r"\^(?P<word>[a-яӧāēйӯ|ᴴ]+)\/(?P<parse>.*)\
$", file)
      for w in words:
```

```
        if w not in seen:
          guesses = guess_word_acc(w[0])
          if len(guesses) != 0:
            if w[1] in guesses:
                fully_guessed += 1
            #else:
                #print('FAIL')
                #print('standard:', w[0], w[1])
                #print(guesses)
            guessed_tags = [re.findall(r"<.*>", l)[0] for l in
guesses]
            true_tags = re.findall(r"<.*>", w[1])[0]
            if true_tags in guessed_tags:
              tags_guessed += 1
            recog += 1
          total +=1
          seen.append(w)
      print('total:', total)
      print('recognised:', recog)
      print('types recognised:', recog/total)
      print('fully_guessed:', fully_guessed)
      print('fully_guessed_all:', fully_guessed / total)
      print('fully_guessed_recog:', fully_guessed / recog)
      print('tags_guessed:', tags_guessed / total)

check_guesser_acc('text-1-gold.txt')
```

```
total: 76.0
recognised: 64.0
types recognised: 0.8421052631578947
fully_guessed: 26.0
fully_guessed_all: 0.34210526315789475
fully_guessed_recog: 0.40625
tags_guessed: 0.34210526315789475
```

```
check_guesser_acc('text-2-gold.txt')
```

```
total: 79.0
recognised: 68.0
types recognised: 0.8607594936708861
fully_guessed: 23.0
fully_guessed_all: 0.2911392405063291
fully_guessed_recog: 0.3382352941176471
tags_guessed: 0.2911392405063291
```

Perfect tokens

```
def check_guesser_acc_perfect(path):
  with open(path, 'r', encoding='utf-8') as file:
      file = file.read()
```

```
      fully_guessed = 0.0
      recog = 0.0
      total = 0.0
      tags_guessed = 0.0
      words = re.findall(r"\^(?P<word>[a-яōāēīūʸʰ]+)\/(.*<(n|v|num|
dem)>.*)\$", file)
      seen = []
      for w in words:
        if w not in seen:
          guesses = guess_word_acc(w[0])
          if len(guesses) != 0:
            if w[1] in guesses:
                fully_guessed += 1
            #else:
               #print('FAIL')
               #print('standard:', w[0], w[1])
               #print(guesses)
            guessed_tags = [re.findall(r"<.*>", l)[0] for l in
guesses]
            true_tags = re.findall(r"<.*>", w[1])[0]
            if true_tags in guessed_tags:
              tags_guessed += 1
            recog += 1
          total +=1
          seen.append(w)

      print('total:', total)
      print('recognised:', recog)
      print('types recognised:', recog/total)
      print('fully_guessed:', fully_guessed)
      print('fully_guessed_all:', fully_guessed / total)
      print('fully_guessed_recog:', fully_guessed / recog)
      print('tags_guessed:', tags_guessed / total)

check_guesser_acc_perfect('text-1-gold.txt')
```

```
total: 52.0
recognised: 46.0
types recognised: 0.8846153846153846
fully_guessed: 26.0
fully_guessed_all: 0.5
fully_guessed_recog: 0.5652173913043478
tags_guessed: 0.5
```

```
check_guesser_acc_perfect('text-2-gold.txt')
```

```
total: 51.0
recognised: 46.0
types recognised: 0.9019607843137255
fully_guessed: 23.0
```

```
fully_guessed_all: 0.45098039215686275
fully_guessed_recog: 0.5
tags_guessed: 0.45098039215686275

check_guesser_acc_perfect('text-2-gold.txt')

total: 51.0
recognised: 46.0
types recognised: 0.9019607843137255
fully_guessed: 23.0
fully_guessed_all: 0.45098039215686275
fully_guessed_recog: 0.5
tags_guessed: 0.45098039215686275
```

Verbs & nouns separately

```
def check_guesser_acc_nouns(path):
  with open(path, 'r', encoding='utf-8') as file:
      file = file.read()
      fully_guessed = 0.0
      recog = 0.0
      total = 0.0
      tags_guessed = 0.0
      words = re.findall(r"\^(?P<word>[a-яōāēīūӯ|ᴴ]+)\/(.*<(n)>.*)\$",
file)
      seen = []
      for w in words:
        if w not in seen:
          print(w)
          guesses = guess_word_acc(w[0])
          if len(guesses) != 0:
            if w[1] in guesses:
                fully_guessed += 1
            #else:
               #print('FAIL')
               #print('standard:', w[0], w[1])
               #print(guesses)
            guessed_tags = [re.findall(r"<.*>", l)[0] for l in
guesses]
            true_tags = re.findall(r"<.*>", w[1])[0]
            if true_tags in guessed_tags:
              tags_guessed += 1
            recog += 1
          total +=1
          seen.append(w)

      print('total:', total)
      print('recognised:', recog)
      print('types recognised:', recog/total)
      print('fully_guessed:', fully_guessed)
```

```
      print('fully_guessed_all:', fully_guessed / total)
      print('fully_guessed_recog:', fully_guessed / recog)
      print('tags_guessed:', tags_guessed / total)

check_guesser_acc_nouns('text-1-gold.txt')
```

```
('базайййагъой', 'базай<n><obl><cum><ess>', 'n')
('бикълабашейоьлъи', 'бикълабашейоьлъи<n>', 'n')
('л|и', 'л|и<n><abs>', 'n')
('аьл|аьъаьш', 'аьл|<n><obl><in><ess><abl>', 'n')
('миц', 'миц<n><abs>', 'n')
('бикълабашейолъил|она', 'бикълабашейолъи<n><abs><quot><add>', 'n')
('ъаьʰгъаь', 'ъаьʰгъаь<n><abs>', 'n')
('бикълабашейолъина', 'бикълабашейолъи<n><abs><add>', 'n')
('гьекьар', 'гьекьар<n><abs>', 'n')
('оьлоьхъаьн', 'оьлоьхъаьн<n><abs>', 'n')
('к|ималид', 'к|има<n><obl><ins>', 'n')
('к|има', 'к|има<n><abs>', 'n')
('ц|ит|ад', 'ц|ит|<n><obl><ins>', 'n')
('роьъил', 'роьъил<n><abs>', 'n')
('инкар', 'инкар<n><abs>', 'n')
('къимат', 'къимат<n><abs>', 'n')
('гемо', 'гемо<n><abs>', 'n')
('гьикматна', 'гьикмат<n><abs><add>', 'n')
total: 18.0
recognised: 12.0
types recognised: 0.6666666666666666
fully_guessed: 3.0
fully_guessed_all: 0.16666666666666666
fully_guessed_recog: 0.25
tags_guessed: 0.16666666666666666
```

```
check_guesser_acc_nouns('text-2-gold.txt')
```

```
('дунналнакъодā', 'дуннал<n><abs><add><irr>', 'n')
('аллагь', 'аллагь<n><abs>', 'n')
('интернетна', 'интернет<n><abs><conj>', 'n')
('к|ет|ослъина', 'к|ет|ослъи<n><abs><add>', 'n')
('телевизорлана', 'телевизор<n><obl><erg><add>', 'n')
('ийо', 'ийо<n><abs>', 'n')
('аболкъодā', 'або<n><obl><dat><irr>', 'n')
('руслан', 'руслан<n><abs>', 'n')
('оьмроь', 'оьмроь<n><abs>', 'n')
('эркенлъина', 'эркенлъи<n><abs><add>', 'n')
('або', 'або<n><abs>', 'n')
('загьматлъи', 'загьмат<n><abs>', 'n')
('хунзахъ', 'хунзахъ<n><abs>', 'n')
('жо', 'жо<n><abs>', 'n')
('оьмроьлаькьаь', 'оьмроь<n><obl><sup><ess>', 'n')
('оьмроьнаь', 'оьмроь<n><abs><add>', 'n')
```

```
('абона', 'або<n><abs><add>', 'n')
('аьгаьрлъина', 'аьгаьрлъи<n><abs><add>', 'n')
('заманна', 'заман<n><abs><add>', 'n')
('телевизорлиъ', 'телевизор<n><obl><in><ess>', 'n')
('интернетлиъ', 'интернет<n><obl><in><ess>', 'n')
('аьгаьрлъли', 'аьгаьрлъли<n><abs>', 'n')
total: 22.0
recognised: 19.0
types recognised: 0.8636363636363636
fully_guessed: 3.0
fully_guessed_all: 0.13636363636363635
fully_guessed_recog: 0.15789473684210525
tags_guessed: 0.13636363636363635
```

```
def check_guesser_acc_verbs(path):
  with open(path, 'r', encoding='utf-8') as file:
      file = file.read()
      fully_guessed = 0.0
      recog = 0.0
      total = 0.0
      tags_guessed = 0.0
      words = re.findall(r"\^(?P<word>[a-яӑӓёйӳӏʜ]+)\/(.*<(v)>.*)\$",
file)
      seen = []
      for w in words:
        if w not in seen:
          guesses = guess_word_acc(w[0])
          if len(guesses) != 0:
            if w[1] in guesses:
                fully_guessed += 1
            else:
               print('FAIL')
               print('standard:', w[0], w[1])
               print(guesses)
            guessed_tags = [re.findall(r"<.*>", l)[0] for l in
guesses]
            true_tags = re.findall(r"<.*>", w[1])[0]
            if true_tags in guessed_tags:
              tags_guessed += 1
            recog += 1
          total +=1
          seen.append(w)

      print('total:', total)
      print('recognised:', recog)
      print('types recognised:', recog/total)
      print('fully_guessed:', fully_guessed)
      print('fully_guessed_all:', fully_guessed / total)
      print('fully_guessed_recog:', fully_guessed / recog)
      print('tags_guessed:', tags_guessed / total)
```

```
check_guesser_acc_verbs('text-1-gold.txt')

FAIL
standard: йуьч|йаьгъеч|е <IV>уьч|<IV>аьгъ<v><neg><pfv.cvb>
['йуьч|<antip><pl>гъо<v><neg><pfv.cvb>',
'йуьч|<antip><pl>гъ<v><neg><pfv.cvb>', 'йуьч|йаьгъо<v><neg><pfv.cvb>',
'йуьч|йаьгъе<v><neg><pfv.cvb>', 'йуьч|йаьгъи<v><neg><pfv.cvb>',
'йуьч|йаьгъоь<v><neg><pfv.cvb>', 'йуьч|йаьгъа<v><neg><pfv.cvb>']
FAIL
standard: богьлъол <III>ов<v><ant.cvb>
['б<m>гь<v><ant.cvb>', 'бог<v><ant.cvb>', 'богь<n><obl><dat>',
'богьлъо<n><obl><dat>', 'богьл<n><in><ess><lat>',
'богьл<n><obl><in><ess><lat>', 'богьлъ<n><obl><dat>',
'богьлхъ<n><obl><dat>', 'богьлъ<n><dat>', 'богьлъ<obl><dat>',
'богьлъоа<n><obl><dat>', 'богьлъое<n><obl><dat>',
'богьлъонзил<n><obl><dat>', 'богьлъоам<n><obl><dat>',
'богьлъоо<n><obl><dat>', 'богьлъохāь<n><obl><dat>',
'богьлъоᴴ<n><obl><dat>']
FAIL
standard: йуьч|на <II>уьч|<v><pfv.cvb>
['йуьч|на<n><pl><abs>', 'йуьч|<v><pfv.cvb>', 'йуьч|ни<n><pl><abs>',
'йуьч<num><dstr><abs>', 'йуьча<num><dstr><abs>', 'йуьч|<num><part>',
'йуьч|н<n><pl><abs>', 'йуьч|наь<n><pl><abs>', 'йуьч|на<v><imp>',
'йуьч|не<v><imp>', 'йуьч|ни<v><imp>', 'йуьч|но<v><imp>',
'йуьч|не<n><pl><abs>', 'йуьч|но<n><pl><abs>', 'йуьч|ноь<n><pl><abs>']
FAIL
standard: бойч|е <III>ов<v><neg><pfv.cvb>
['бо<v><neg><pfv.cvb>']
FAIL
standard: йиᴴъилна <IV>иᴴъил<v><pfv.cvb>
['йиᴴъил<num><part>', 'йиᴴъил<v><pfv.cvb>', 'йиᴴъилна<n><pl><abs>',
'йиᴴъилни<n><pl><abs>', 'йиᴴъилн<n><pl><abs>',
'йиᴴъилнаь<n><pl><abs>', 'йиᴴъил<dstr><abs>',
'йиᴴъил<num><dstr><abs>', 'йиᴴъилна<v><imp>', 'йиᴴъилне<v><imp>',
'йиᴴъилни<v><imp>', 'йиᴴъилно<v><imp>', 'йиᴴъилне<n><pl><abs>',
'йиᴴъилно<n><pl><abs>', 'йиᴴъилноь<n><pl><abs>']
FAIL
standard: йиᴴъилйугъолъ <IV>иᴴъил<IV>угъо<v><simul.cvb>
['йиᴴъил<II>угъо<v><simul.cvb>', 'йиᴴъил<IV>угъо<v><simul.cvb>',
'йиᴴъил<nhpl>угъо<v><simul.cvb>', 'йиᴴъилй<I>угъо<v><simul.cvb>',
'йиᴴъилйугъо<n><obl><cont><ess>', 'йиᴴъилйуг<obl><cont><ess>',
'йиᴴъилйугъ<n><obl><cont><ess>', 'йиᴴъилйугхъ<n><obl><cont><ess>',
'йиᴴъилйугъ<n><cont><ess>', 'йиᴴъилйугъ<obl><cont><ess>',
'йиᴴъилйугъоа<n><obl><cont><ess>', 'йиᴴъилйугъое<n><obl><cont><ess>',
'йиᴴъилйугъонзил<n><obl><cont><ess>',
'йиᴴъилйугъоам<n><obl><cont><ess>', 'йиᴴъилйугъоо<n><obl><cont><ess>',
'йиᴴъилйугъохāь<n><obl><cont><ess>',
'йиᴴъилйугъоᴴ<n><obl><cont><ess>', 'йиᴴъилйугъо<n><in><ess>',
'йиᴴъилйугъола<n><obl><in><ess>', 'йиᴴъилйугъоле<n><obl><in><ess>',
'йиᴴъилйугъолнзил<n><obl><in><ess>',
```

```
'йиᴴъилйугъолам<n><obl><in><ess>', 'йиᴴъилйугъоло<n><obl><in><ess>',
'йиᴴъилйугъолхāь<n><obl><in><ess>', 'йиᴴъилйугъолᴴ<n><obl><in><ess>']
FAIL
standard: бāхъна <hpl>a<pl>хъ<v><pfv.cvb>
['бāхъ<v><pfv.cvb>', 'бāхъна<n><pl><abs>', 'бāх<n><in><ess><add>',
'бāха<n><obl><in><ess><add>', 'бāхе<n><obl><in><ess><add>',
'бāхнзил<n><obl><in><ess><add>', 'бāхам<n><obl><in><ess><add>',
'бāхо<n><obl><in><ess><add>', 'бāххāь<n><obl><in><ess><add>',
'бāхᴴ<n><obl><in><ess><add>', 'бāхъни<n><pl><abs>',
'бāхънаь<n><pl><abs>', 'бāха<num><dstr><abs>', 'бāхъ<num><part>',
'бāхън<n><pl><abs>', 'бāхъ<dstr><abs>', 'бāхъна<v><imp>',
'бāхъне<v><imp>', 'бāхъни<v><imp>', 'бāхъно<v><imp>',
'бāхъне<n><pl><abs>', 'бāхъно<n><pl><abs>', 'бāхъноь<n><pl><abs>']
total: 29.0
recognised: 29.0
types recognised: 1.0
fully_guessed: 22.0
fully_guessed_all: 0.7586206896551724
fully_guessed_recog: 0.7586206896551724
tags_guessed: 0.7586206896551724

check_guesser_acc_verbs('text-2-gold.txt')

FAIL
standard: йовал <IV>ов<v><inf>
['йо<v><antip><inf>', 'йо<v><pl><inf>', 'йо<v><pst.ptsp><obl><dat>',
'йо<n><pl><obl><dat>', 'йо<n><obl><dat>', 'й<m>в<v><inf>',
'йова<n><pl><obl><dat>', 'йова<n><obl><dat>', 'йо<antip><v><inf>',
'йо<pl><v><inf>', 'йоᴴ<n><pl><obl><dat>', 'йов<n><pl><obl><dat>',
'йови<n><pl><obl><dat>', 'йоваь<n><pl><obl><dat>',
'йово<n><obl><dat>', 'йова<v><inf>', 'йове<v><inf>', 'йови<v><inf>',
'йово<v><inf>', 'йове<n><pl><obl><dat>', 'йово<n><pl><obl><dat>',
'йовоь<n><pl><obl><dat>', 'йов<dat>', 'йов<n><obl><dat>',
'йовче<n><obl><dat>', 'йованзил<n><obl><dat>', 'йовйо<n><obl><dat>',
'йова<n><dat>', 'йоваа<n><obl><dat>', 'йоваам<n><obl><dat>',
'йовахāь<n><obl><dat>', 'йоваᴴ<n><obl><dat>']
FAIL
standard: йегāйо <IV>егā<v><pst>
['йегā<v><pst>']
FAIL
standard: бегākъала <III>ега<v><neg.opt>
['<III>ега<v><proh><conj>', '<hpl>ега<v><proh><conj>',
'беге<v><proh><conj>', 'беги<v><proh><conj>', 'бего<v><proh><conj>',
'бегоь<v><proh><conj>', 'бегāᴴкъе<v><imp><conj>',
'бегāᴴкъо<v><imp><conj>', 'бегākъа<n><pl><abs><conj>',
'бегākъо<n><pl><abs><conj>', 'бегākъ<n><pl><abs><conj>',
'бегākъаь<n><pl><abs><conj>', 'бегākъа<n><pl><obl><genII>',
'бегākъо<n><pl><obl><genII>', 'бегākъ<n><pl><obl><genII>',
'бегākъаь<n><pl><obl><genII>', 'бегākъо<n><obl><genII>',
'бегākъ<n><obl><genII>', 'бегāᴴкъе<v><imp><prmI>',
```

'бегāᴴкъо<v><imp><prmI>', 'бегāкъа<n><pl><abs>',
'бегāкъала<n><pl><abs>', 'бегāкъали<n><pl><abs>',
'бегāкъ<v><imp><conj>', 'бегāкъи<n><pl><abs><conj>',
'бегāкъи<n><pl><obl><genII>', 'бегāкъа<n><obl><genII>',
'бегāк<n><obl><genII>', 'бегāкъ<v><imp><prmI>',
'бегāкᴴъа<antip><v><imp>', 'бегāкᴴъал<v><imp>',
'бегāкъа<v><imp><conj>', 'бегāкъи<v><imp><conj>',
'бегāкъе<n><pl><abs><conj>', 'бегāкъоь<n><pl><abs><conj>',
'бегāкъе<n><pl><obl><genII>', 'бегāкъоь<n><pl><obl><genII>',
'бегāкъ<genII>', 'бегāкъче<n><obl><genII>',
'бегāкъанзил<n><obl><genII>', 'бегāкъйо<n><obl><genII>',
'бегāкъа<v><imp><prmI>', 'бегāкъи<v><imp><prmI>',
'бегāкъе<antip>ле<v><imp>', 'бегāкъе<pl>ле<v><imp>',
'бегāкъи<antip>л<v><imp>', 'бегāкъи<pl>л<v><imp>',
'бегāкъало<n><pl><abs>', 'бегāкъа<n><genII>',
'бегāкъаа<n><obl><genII>', 'бегāкъаам<n><obl><genII>',
'бегāкъаь<n><obl><genII>', 'бегāкъаᴴ<n><obl><genII>',
'бегāкъал<n><pl><abs>', 'бегāкъалаь<n><pl><abs>', 'бегāкъала<v><imp>',
'бегāкъали<v><imp>', 'бегāкъало<v><imp>', 'бегāкъале<n><pl><abs>',
'бегāкъалоь<n><pl><abs>', 'бегāкъал<dstr><abs>']
total: 20.0
recognised: 20.0
types recognised: 1.0
fully_guessed: 17.0
fully_guessed_all: 0.85
fully_guessed_recog: 0.85
tags_guessed: 0.85