

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
ХЭРЭГЛЭЭНИЙ ШИНЖЛЭХ УХААН, ИНЖЕНЕРЧЛЭЛИЙН СУРГУУЛЬ
МЭДЭЭЛЭЛ, КОМПЬЮТЕРИЙН УХААНЫ ТЭНХИМ

Цэрэнбямбаагийн Хасбилэгт

ГАР БИЧМЭЛ ТАНИЛТАНД ЗОРИУЛСАН
МУИС-Н НЭЭЛТТЭЙ ӨГӨГДӨЛ БЭЛТГЭХ НЬ
(Preparation of open NUM handwritten character dataset)

Мэдээллийн технологи (D061303)
Бакалаврын судалгааны ажил

Улаанбаатар

2020 он

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ
ХЭРЭГЛЭЭНИЙ ШИНЖЛЭХ УХААН, ИНЖЕНЕРЧЛЭЛИЙН СУРГУУЛЬ
МЭДЭЭЛЭЛ, КОМПЬЮТЕРИЙН УХААНЫ ТЭНХИМ

ГАР БИЧМЭЛ ТАНИЛТАНД ЗОРИУЛСАН МУИС-Н
НЭЭЛТТЭЙ ӨГӨГДӨЛ БЭЛТГЭХ НЬ
(Preparation of open NUM handwritten character dataset)

Мэдээллийн технологи (D061303)
Бакалаврын судалгааны ажил

Удирдагч: _____ Др. Б.Сувдаа

Гүйцэтгэсэн: _____ Ц.Хасбилэгт (15B1SEAS0980)

Улаанбаатар

2020 он

Зохиогчийн баталгаа

Миний бие Цэрэнбямбаагийн Хасбилэгт "ТАР БИЧМЭЛ ТАНИЛТАНД ЗОРИУЛСАН МУИС-Н НЭЭЛТТЭЙ ӨГӨГДӨЛ БЭЛТГЭХ НЬ" сэдэвтэй судалгааны ажлыг гүйцэтгэсэн болохыг зарлаж дараах зүйлсийг баталж байна:

- Ажил нь бүхэлдээ эсвэл ихэнхдээ Монгол Улсын Их Сургуулийн зэрэг горилохоор дэвшүүлсэн болно.
- Энэ ажлын аль нэг хэсгийг эсвэл бүхлээр нь ямар нэг их, дээд сургуулийн зэрэг горилохоор оруулж байгаагүй.
- Бусдын хийсэн ажлаас хуулбарлаагүй, ашигласан бол ишлэл, зүүлт хийсэн.
- Ажлыг би өөрөө (хамтарч) хийсэн ба миний хийсэн ажил, үзүүлсэн дэмжлэгийг дипломын ажилд тодорхой тусгасан.
- Ажилд тусалсан бүх эх сурвалжид талархаж байна.

Гарын үсэг: _____

Огноо: _____

Contents

ЗУРГИЙН ЖАГСААЛТ	iii
ХҮСНЭГТИЙН ЖАГСААЛТ	iv
УДИРТГАЛ	1
Зорилго	2
Зорилт	2
БҮЛГҮҮД	3
1. СУДАЛГАА	3
1.1 Бусад хэл дээрх өгөгдлийн сангууд	3
2. АРГЫН ТОДОРХОЙЛОЛТ	4
2.1 Оролт	4
2.2 Боловсруулалт	4
2.3 Гаралт	4
3. ХӨГЖҮҮЛЭЛТ	5
4. ДҮГНЭЛТ	6
НОМ ЗҮЙ	7
ХАВСРАЛТ	8
A. СЭРЭЭХ ҮГ ИЛРҮҮЛЭГЧ	8
B. KALDI GSTREAMER СЕРВЕРТ ЗОРИУЛСАН ТОХИРГООНЫ ФАЙЛ	9
C. ЯРИА ТАНИХ ХЭСЭГ	10
D. ЧАДВАР СОНГОХ	11
E. ЦАГ, ОГНООНЫ МЭДЭЭЛЭЛ АВАХ	13
F. ЯРИА ҮҮСГҮҮР	14

List of Figures

Зураг

Хуудас

List of Tables

Хүснэгт

Хуудас

Кодын жагсаалт

Код

Хуудас

УДИРТГАЛ

Өнөөдөр хиймэл оюун ухаан болон машин сургалтын салбаруудын технологи, судалгаанууд хөгжихийн хэрээр, өмнө нь хүний төсөөлж байгаагүй олон шинэ боломжууд, зөвхөн технологийн салбаруудаар хязгаарлагдалгүй, ар араасаа нээгдэж байгаагаас гадна өнөөдөр нийтэд ашиглагдаж буй технологиудын нарийвчлал, үр дүн, чанар ч мөн үүнийг дагаад сайжирч байгаа билээ. Энэ дундаас гар бичмэл танилттай холбоотой судалгаанууд олноор хөгжүүлэгдэх нь бидний мэдэх хүн компьютерийн харилцааг эергээр өөрчилж байна.

Зорилго

Энэ чиглэлийн судалгаанууд, түүн дээр суурилсан програм хангамж, үйлчилгээ, бүтээгдэхүүн хөгжүүлэгдэхэд гар бичмэлийг таних, машин сургалтанд ашиглах өгөгдөл зайлшгүй хэрэгтэй, мөн Монгол хэл дээр энэ төрлийн нээлттэй, нэгдсэн өгөгдлийн сан хомс учир энэхүү судалгааны ажлаар өөрийн эх хэл дээрх нээлттэй гар бичмэлийн санг бий болгох, санг үүсгэхэд ашиглах арга, хэрэгслийг тодорхойлохыг зорьсон юм.

Зорилт

Монгол хэл дээрх гар бичмэлийн нээлттэй санг үүсгэхдээ дараах үе шатын дагуу ажиллана.

1. Үндсэн шаардлагуудыг тодорхойлох
2. Тэмдэгтүүд бичүүлж авах маягтын загварыг гаргах
3. Бөглөгдсөн маягт дээр боловсруулалт хийх, тэмдэгтүүдийг ялгаж авах
4. Их хэмжээний оролтын үед боловсруулалт хийх боломжтой байдлаар програмыг хөгжүүлэх
5. Програмыг PyPi, GitHub дээр байршуулах, програмтай холбоотой мэдээлэл, ашиглах зааврыг бичих

1. СУДАЛГАА

1.1 Бусад хэл дээрх өгөгдлийн сангууд

1.1.1 *NIST 19 special database*

1.1.2 *MNIST*

1.1.3 *EMNIST*

[2] нь

1.1.4 *IAM*

Машин сургалт тэр дундаа гар бичмэл танилттай холбоотой машин сургалтын өгөгдлийг хэрхэн бэлдэх талаарх судалгааны ажил хомс байсан тул өгөгдлийн хэлбэр, хэмжээ болон хэрхэн зохион байгуулагдсан зэргийг энэ чиглэлийн өөр бусад хэл дээрх түгээмэл ашиглагддаг нээлттэй өгөгдлүүдтэй адил байдлаар зохион байгуулахаар шийдсэн.

2. АРГЫН ТОДОРХОЙЛОЛТ

2.1 Оролт

2.2 Боловсруулалт

2.3 Гаралт

3. ХӨГЖҮҮЛЭЛТ

4. ДҮГНЭЛТ

Энэхүү судалгааны ажлаар бусад хэл дээрх гар бичмэл танилттай холбоотой судалгаануудын ажлууд дээр тулгуурлан өөрийн хэлний гар бичмэл танилтанд зориулсан нээлттэй сан, түүнд ашиглах програм хангамжийг хөгжүүлэхээр зорьлоо. Монгол хэл дээрх гар бичмэл танилт нь оюун ухаан, машин сургалтын салбарын энгийн судалгааны ажил биш бөгөөд бичиг баримтыг дижиталжуулах, хүний бичгийн хэвийг ялган таних (authentication), баталгаажуулах (authorization) гэх мэт гар бичмэлтэй холбоотой асар олон төрлийн ажил, судалгаа, програм хангамжуудад ч мөн ашиглагдах боломжтой. Тийм ч учраас дээрх салбаруудтай холбоотой ажил, судалгаа хийхээр зорьж буй хүмүүст энэхүү судалгааны ажил суурь судалгаа нь болох бөгөөд хамгийн чухал нь Монгол хэлний гар бичмэлтэй холбоотой судалгаануудад ашиглагдах нэгдсэн нэг өгөгдлийн сан болон цаашид тогтмол хөгжүүлэгдэж байх хэрэгтэй.

Нийт өгөгдлийн тоо хэмжээ, төрлүүдийг нэмэх, өргөжүүлэхийн тулд python хэл дээрх програм хангамжийн боломжуудыг нэмэх, бичвэр агуулах маягтыг динамик байдлаар үүсгэх, ажиллагааг хурдасгах, нийтээс маягтын зургаа авч автоматаар боловсруулан сангаа шинэчлэх бие даасан вэбсайт, систем гэх мэт ажлууд нэмэгдэж хийх боломжтой.

Ашигласан материал

- [1] Smart Speaker — Wikipedia, The Free Encyclopedia

https://en.wikipedia.org/wiki/Smart_speaker

- [2] Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017). *EMNIST: an extension of MNIST to handwritten letters*.

<http://arxiv.org/abs/1702.05373>

A. СЭРЭЭХ ҮГ ИЛРҮҮЛЭГЧ

```
1 from precise_runner import PreciseEngine, PreciseRunner
2 import time
3 from pixel_ring import pixel_ring # ReSpeaker led controller
4 import mraa
5 import os
6
7 # (pixel ring)
8 en = mraa.Gpio(12)
9 if os.getuid() != 0:
10     time.sleep(1)
11 en.dir(mraa.DIR_OUT)
12 en.write(0)
13 pixel_ring.set_brightness(20)
14
15 # \ \ \ \
16 def on_activation():
17     print('hello')
18     pixel_ring.wakeup() # \ \ \ \
19     # ...
20     # \ ,\ \ \ \ \ \ \ \ ...
21     # ...
22     time.sleep(3)
23     pixel_ring.off()
24
25 def main():
26     # \ \ \
27     engine = PreciseEngine('precise-engine/precise-engine', 'hey-obi.
28         pb') # hey-obi.pb\ \ \
29     runner = PreciseRunner(engine, on_activation=on_activation)
30     runner.start()
31
32 if __name__ == '__main__':
33     main()
34     while True:
35         time.sleep(1)
```

В. KALDI GSTREAMER СЕРВЕРТ ЗОРИУЛС ТОХИРГООНЫ ФАЙЛ

```
1 timeout-decoder : 10
2 decoder:
3     model: /opt/models/tri3b/final.mdl
4     lda-mat: /opt/models/tri3b/final.mat
5     word-syms: /opt/models/tri3b/graph/words.txt
6     fst: /opt/models/tri3b/graph/HCLG.fst
7     silence-phones: "1:2:3:4:5"
8     left-context: 3
9     right-context: 3
10 out-dir: tmp
11
12 use-vad: False
13 silence-timeout: 5
14
15 ...
```


С. ЯРИА ТАНИХ ХЭСЭГ

```
1 def stt():
2     audio_file = '.tmp/stt.flac'
3     text = False
4     # \ \
5     subprocess.run(['rec', '-c', '1', '-r', '16000', '-d', audio_file
6                     ,
7                     'trim', '0', '15', 'silence', '1', '0.1', '0.3%',
8                     '1', '3.0', '0.3%'])
9     # Curl\ \ \ HTTP \
10    output = str(subprocess.check_output(
11        "curl -T {} {}".format(audio_file, conf.config()['STT_URL']),
12        shell=True), 'utf-8')
13    print(output)
14    if output == 'No workers available':
15        return False
16    response = json.loads(output)
17    print(response)
18    # \ \ \ \ \ \ \ \ \ \
19    if response['status'] == 0:
20        text = response['hypotheses'][0]['utterance']
21    return text
```

D. ЧАДВАР СОНГОХ

```
1 result = {'obj': None, 'answer': ''}
2 if intent['intent']['confidence'] > 0.5:
3     # intent \ \
4     #
5     if intent['intent']['name'] == 'greeting':
6         result['answer'] = greeting.greet()
7     #
8     elif intent['intent']['name'] == 'get_time':
9         result['answer'] = time_n_date.get_time()
10    #
11    elif intent['intent']['name'] == 'get_date':
12        result['answer'] = time_n_date.get_date()
13    # \ \ \ \
14    elif intent['intent']['name'] == 'get_day':
15        result['answer'] = time_n_date.get_day()
16    # \
17    elif intent['intent']['name'] == 'get_weather':
18        p = {}
19        # entity \ \ \ \ \ \ \ \
20        for entity in intent['entities']:
21            p[entity['entity']] = entity['value']
22        result['answer'] = weather.get_weather(**p)
23    # \
24    elif intent['intent']['name'] == 'get_currency':
25        if len(intent['entities']) > 0:
26            result['answer'] = currency.get_currency(
27                intent['entities'][0]['value'])
28        else:
29            result['answer'] = currency.get_currency()
30    #
31    elif intent['intent']['name'] == 'order':
32        if len(intent['entities']) > 0:
33            result['obj'] = order.Order(intent['entities'][0]['value']
34                ])
35        else:
36            result['obj'] = order.Order()
37            result['answer'] = ' {} '.format(
38                result['obj']._name)
39    elif intent['intent']['name'] == 'order_size':
40        if len(intent['entities']) > 0:
41            result['obj'] = obj.set_size(intent['entities'][0]['value']
42                ])
43        else:
44            result['obj'] = obj.set_size()
45            result['answer'] = ' '
46    elif intent['intent']['name'] == 'order_quantity':
47        if len(intent['entities']) > 0:
48            obj.set_quantity(intent['entities'][0]['value'])
49        else:
50            obj.set_quantity()
51            result['answer'] = obj.order()
52    else:
```

```
51     result = False
52 #     \     \     \     \     \     \     \     \
53 if result == False:
54     result = {'answer': DO_NOT_KNOW[math.floor(
55         random.random() * len(DO_NOT_KNOW))], 'obj': obj}
56 return result
```

Е. ЦАГ, ОГНООНЫ МЭДЭЭЛЭЛ АВАХ

```
1 from datetime import date, datetime
2
3 WEEKDAY_NAMES = [' ', ' ', ' ', ' ', ' ', ' ', ' ']
4
5 def get_time():
6     now = datetime.now()
7     hour = now.hour
8     minute = now.minute
9     return str(hour) + ' ' + str(minute) + ' '
10
11 def get_day():
12     today = date.today()
13     day = WEEKDAY_NAMES[today.weekday()]
14     return ' ' + day + ' '
15
16 def get_date():
17     today = date.today()
18     month = today.month
19     day = today.day
20     return ' ' + str(month) + ' ' + str(day) + ' - '
```

Б. ЯРИА ҮҮСГҮҮР

```
1 # \ \ \ \
2 def get_wav(sentence=''):
3     url = config()['TTS_URL'] + str(sentence)
4     r = requests.get(url)
5     with open('.tmp/tts.wav', 'wb') as f:
6         f.write(r.content)
7     return './.tmp/tts.wav'
8
9 # \ \
10 def play_wav(location='.tmp/tts.wav'):
11     chunk = 1024
12     f = wave.open(location, "rb")
13     p = pyaudio.PyAudio()
14     stream = p.open(format=p.get_format_from_width(f.getsampwidth()),
15                     channels=f.getnchannels(), rate=f.getframerate(), output=True
16                     )
17     data = f.readframes(chunk)
18     while data:
19         stream.write(data)
20         data = f.readframes(chunk)
21     stream.stop_stream()
22     stream.close()
23     p.terminate()
```