Орчиноо бэлдэх

stanfordnlp болон stanza татна. stanfordnlp нь pytorch 1.4.0 дээр ажилладаг учраас pytorch 1.4.0 суулгана.

```
!pip install stanfordnlp
!pip install stanza
```

```
Requirement already satisfied: stanfordnlp in /usr/local/lib/python3.7/dist-packages (0
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from star
Requirement already satisfied: protobuf in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from sta
Requirement already satisfied: torch>=1.0.0 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: stanza in /usr/local/lib/python3.7/dist-packages (1.3.0)
Requirement already satisfied: protobuf in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from sta
Requirement already satisfied: emoji in /usr/local/lib/python3.7/dist-packages (from sta
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from stanz
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from star
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: torch>=1.3.0 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packa
```

```
!pip install --pre torch torchvision -f https://download.pytorch.org/whl/nightly/cu102/torch_
```

```
Looking in links: https://download.pytorch.org/whl/nightly/cu102/torch_nightly.html
Requirement already satisfied: torch in /usr/local/lib/python3.7/dist-packages (1.10.0+c
Collecting torch
  Downloading https://download.pytorch.org/whl/nightly/cu102/torch-1.12.0.dev20220413%2B
     |████████████████████████████████| 741.7 MB 8.8 kB/s
Requirement already satisfied: torchvision in /usr/local/lib/python3.7/dist-packages (0
Collecting torchvision
  Downloading https://download.pytorch.org/whl/nightly/cu102/torchvision-0.13.0.dev2022
     |████████████████████████████████| 19.5 MB 1.3 MB/s
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from tor
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (f
```

```
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib
Installing collected packages: torch, torchvision
  Attempting uninstall: torch
    Found existing installation: torch 1.10.0+cu111
    Uninstalling torch-1.10.0+cu111:
      Successfully uninstalled torch-1.10.0+cu111
```

```
!pip install torch==version
!pip install torch==1.4.0
```

```
ERROR: Could not find a version that satisfies the requirement torch==version (from vers
ERROR: No matching distribution found for torch==version
Collecting torch==1.4.0
  Using cached torch-1.4.0-cp37-cp37m-manylinux1_x86_64.whl (753.4 MB)
Installing collected packages: torch
  Attempting uninstall: torch
    Found existing installation: torch 1.12.0.dev20220413+cu102
    Uninstalling torch-1.12.0.dev20220413+cu102:
      Successfully uninstalled torch-1.12.0.dev20220413+cu102
ERROR: pip's dependency resolver does not currently take into account all the packages t
torchvision 0.13.0.dev20220413+cu102 requires torch==1.12.0.dev20220413+cu102, but you h
torchtext 0.11.0 requires torch==1.10.0, but you have torch 1.4.0 which is incompatible
torchaudio 0.10.0+cu111 requires torch==1.10.0, but you have torch 1.4.0 which is incomp
Successfully installed torch-1.4.0
```

Германи хэл сонгоно.

1. Германи хэл татна:stanfordnlp.download('de')
2. pipeline-аа Германи хэл дээр сонгоно.

```
import stanfordnlp
stanfordnlp.download('de')
nlp = stanfordnlp.Pipeline(lang="de")
```

```
Using the default treebank "de_gsd" for language "de".
Would you like to download the models for: de_gsd now? (Y/n)
y

Default download directory: /root/stanfordnlp_resources
Hit enter to continue or type an alternate directory.


Downloading models for: de_gsd
Download location: /root/stanfordnlp_resources/de_gsd_models.zip
```

```
100%|██████████| 229M/229M [00:39<00:00, 5.75MB/s]

Download complete.  Models saved to: /root/stanfordnlp_resources/de_gsd_models.zip
Extracting models file for: de_gsd
Cleaning up...Done.
Use device: cpu
---
Loading: tokenize
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_tokenizer.pt', 'lang':
---
Loading: mwt
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_mwt_expander.pt', 'lang
Building an attentional Seq2Seq model...
Using a Bi-LSTM encoder
Using soft attention for LSTM.
Finetune all embeddings.
---
Loading: pos
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_tagger.pt', 'pretrain_p
---
Loading: lemma
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_lemmatizer.pt', 'lang'
Building an attentional Seq2Seq model...
Using a Bi-LSTM encoder
Using soft attention for LSTM.
Finetune all embeddings.
[Running seq2seq lemmatizer with edit classifier]
---
Loading: depparse
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_parser.pt', 'pretrain_p
Done loading processors!
---
```

Токенчлох, сегментлэх

processors='tokenize' гэж токенчлох үйлдэл хийх процесс сонгоно. sentence.tokens гэж токенруугаа хандана.

```
nlp = stanfordnlp.Pipeline(processors='tokenize', lang='de')
doc = nlp("Ich wohne in der Mongolei. Anuzhin spielt Fußball. Sie kann Russisch")
for i, sentence in enumerate(doc.sentences):
    print(f"====== Sentence {i+1} tokens =======")
····print(*[f"index:·{token.index.rjust(3)}\ttoken:·{token.text}"·for·token·in·sentence.token
```

```
Use device: cpu
---
Loading: tokenize
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_tokenizer.pt', 'lang':
Done loading processors!
---
====== Sentence 1 tokens =======
index:   1      token: Ich
index:   2      token: wohne
index:   3      token: in
index:   4      token: der
index:   5      token: Mongolei
index:   6      token: .
====== Sentence 2 tokens =======
index:   1      token: Anuzhin
index:   2      token: spielt
index:   3      token: Fußball
index:   4      token: .
====== Sentence 3 tokens =======
index:   1      token: Sie
index:   2      token: kann
index:   3      token: Russisch
```

Леммачлал

Леммачлах үйлдэл болох processors='tokenize,mwt,pos,lemma'. word.lemma гэж леммадаа хандана.

```
nlp = stanfordnlp.Pipeline(processors='tokenize,mwt,pos,lemma', lang = 'de')
doc = nlp("Ich wohne in der Mongolei. Anuzhin spielt Fußball. Sie kann Russisch.")
print(*[f'word: {word.text+" "}\tlemma: {word.lemma}' for sent in doc.sentences for word in s
```

```
Use device: cpu
---
Loading: tokenize
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_tokenizer.pt', 'lang':
---
Loading: mwt
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_mwt_expander.pt', 'lang
Building an attentional Seq2Seq model...
Using a Bi-LSTM encoder
Using soft attention for LSTM.
Finetune all embeddings.
---
Loading: pos
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_tagger.pt', 'pretrain_p
---
```

```
Loading: lemma
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_lemmatizer.pt', 'lang'
Building an attentional Seq2Seq model...
Using a Bi-LSTM encoder
Using soft attention for LSTM.
Finetune all embeddings.
[Running seq2seq lemmatizer with edit classifier]
Done loading processors!
---
word: Ich       lemma: ich
word: wohne     lemma: wohnen
word: in        lemma: in
word: der       lemma: der
word: Mongolei  lemma: Mongolei
word: .         lemma: .
word: Anuzhin   lemma: anuzhin
word: spielt    lemma: spielen
word: Fußball   lemma: Fußball
word: .         lemma: .
word: Sie       lemma: Sie|sie
word: kann      lemma: können
word: Russisch  lemma: Russisch
word: .         lemma: .
```

Үгсийн аймаг

Үгсийн аймаг процессор сонгох: processors='tokenize,mwt,pos' Үгсийн аймагруу word.upos-р хандана.

```
nlp = stanfordnlp.Pipeline(processors='tokenize,mwt,pos', lang = 'de')
doc = nlp("Ich lebe in der Mongolei. Google ist eine Organisation")
print(*[f'word: {word.text+" "}\tupos: {word.upos}\txpos: {word.xpos}' for sent in doc.senten
```

```
Use device: cpu
---
Loading: tokenize
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_tokenizer.pt', 'lang':
---
Loading: mwt
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_mwt_expander.pt', 'lang
Building an attentional Seq2Seq model...
Using a Bi-LSTM encoder
Using soft attention for LSTM.
Finetune all embeddings.
---
Loading: pos
With settings:
{'model_path': '/root/stanfordnlp_resources/de_gsd_models/de_gsd_tagger.pt', 'pretrain_p
```

```
Done loading processors!
---
word: Ich       upos: PRON      xpos: PPER
word: lebe      upos: VERB      xpos: VVFIN
word: in        upos: ADP       xpos: APPR
word: der       upos: DET       xpos: ART
word: Mongolei  upos: PROPN     xpos: NE
word: .         upos: PUNCT     xpos: $.
word: Google    upos: PROPN     xpos: NE
word: ist       upos: AUX       xpos: VAFIN
word: eine      upos: DET       xpos: ART
word: Organisation      upos: NOUN      xpos: NN
```

Нэрлэсэн нэр тодорхойлох

Name of entity процессор сонгох: processors='tokenize,ner' Name of entity -н төрөл рүү {ent.type} гэж хандана.

```
import stanza
nlp = stanza.Pipeline(lang='de', processors='tokenize,ner')
doc = nlp("Ich lebe in der Mongolei")
print(*[f'entity: {ent.text}\ttype: {ent.type}' for ent in doc.ents], sep='\n')
```

```
2022-04-14 06:54:25 WARNING: Language de package default expects mwt, which has been add
2022-04-14 06:54:25 INFO: Loading these models for language: de (German):
=======================
| Processor | Package |
-----------------------
| tokenize  | gsd     |
| mwt       | gsd     |
| ner       | conll03 |
=======================

2022-04-14 06:54:25 INFO: Use device: cpu
2022-04-14 06:54:25 INFO: Loading: tokenize
2022-04-14 06:54:25 INFO: Loading: mwt
2022-04-14 06:54:25 INFO: Loading: ner
2022-04-14 06:54:27 INFO: Done loading processors!
entity: Mongolei        type: LOC
```

Өгүүлбэрийн хандлага

```
stanza.download("de")
```

```
Downloading https://raw.githubusercontent.com/stanfordnlp/stanza-
resources/main/resources_1.3.0.json:
2022-04-14 06:49:24 INFO: Downloading default packages for language: de (German)...
Downloading https://huggingface.co/stanfordnlp/stanza-de/resolve/v1.3.0/models/default.zip:
100%
2022-04-14 06:49:41 INFO: Finished downloading models and saved to /root/stanza resource
```

Өгүүлбэрийн хандлага олох процессор сонгох: processors='tokenize,sentiment'

Хандлага-руу хандана: sentence.sentiment

```
nlp = stanza.Pipeline(lang='de', processors='tokenize,sentiment')
doc = nlp('Ich liebe dich. Ich hasse diesen Film. Ich kenne diesen Film. Das macht Spaß!')
for i, sentence in enumerate(doc.sentences):
··if·sentence.sentiment·==·0:
····print(i,·":·Negative")
··elif·sentence.sentiment·==·1:
····print(i,·":·Neutral")
··else:
    print(i, ": Positive")
```

```
2022-04-14 07:09:32 WARNING: Language de package default expects mwt, which has been add
2022-04-14 07:09:32 INFO: Loading these models for language: de (German):
=======================
| Processor | Package |
-----------------------
| tokenize  | gsd     |
| mwt       | gsd     |
| sentiment | sb10k   |
=======================

2022-04-14 07:09:32 INFO: Use device: cpu
2022-04-14 07:09:32 INFO: Loading: tokenize
2022-04-14 07:09:32 INFO: Loading: mwt
2022-04-14 07:09:32 INFO: Loading: sentiment
2022-04-14 07:09:32 INFO: Done loading processors!
0 : Positive
1 : Negative
2 : Neutral
3 : Positive
```

✓ 8s completed at 3:19 PM