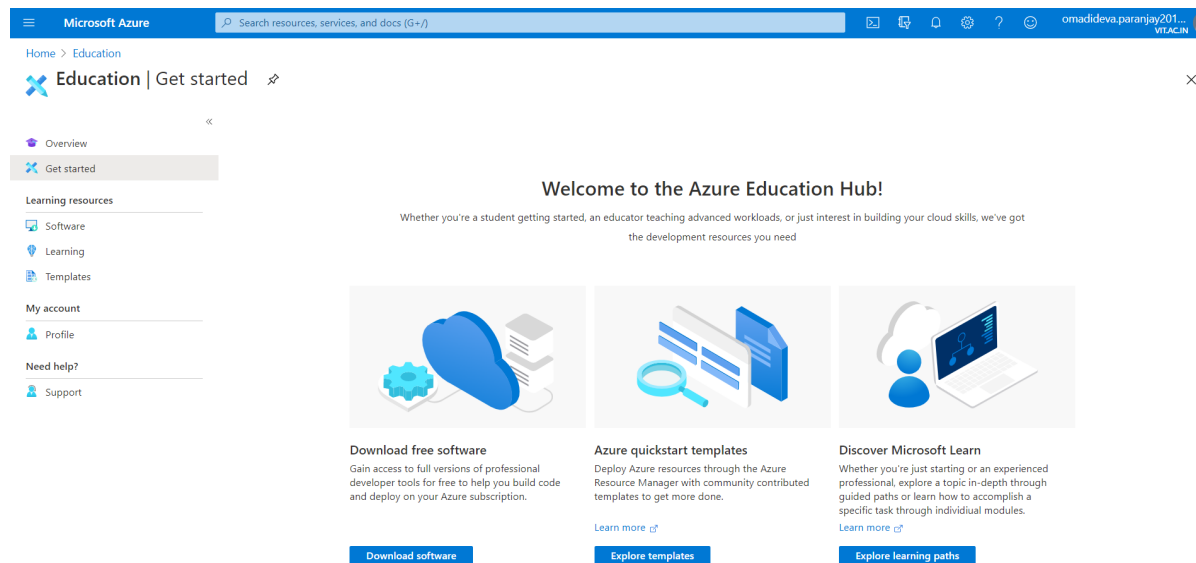


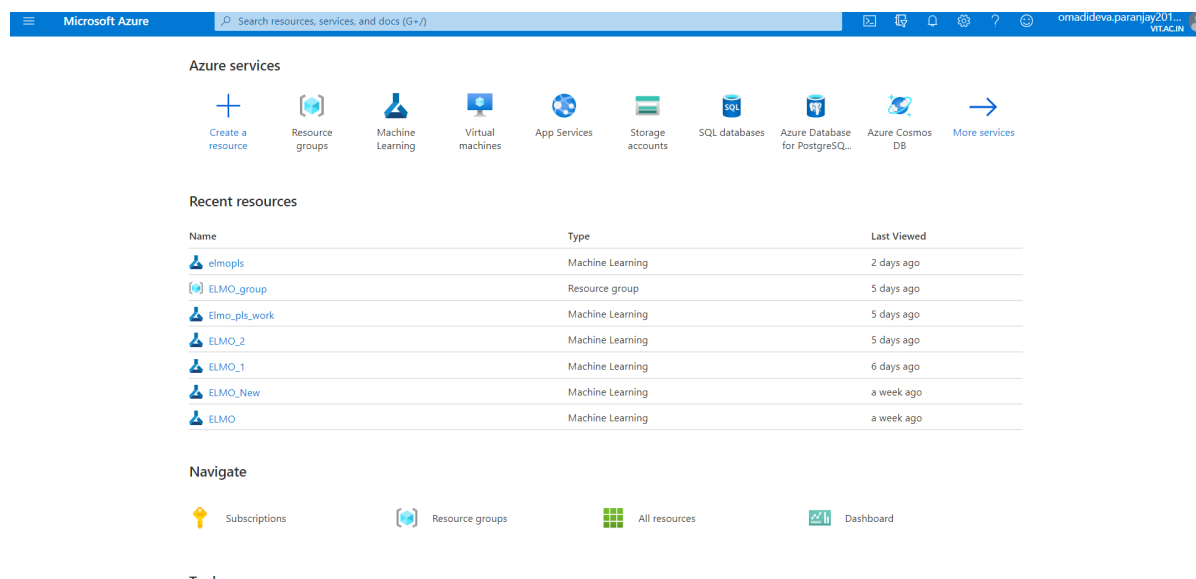
Setting up Azure

The ELMO model was trained on a virtual machine on Microsoft Azure. The first part of the document talks about the setting up procedure for Azure. Here , we assume a Student Developer Pack is used, while other Pay-as-you-go subscriptions can also be opted for.

To start off Google search Azure for Students. The sign up procedure is simple. It would help if you already have a Student Developer Pack activated on your GitHub account , although this is not necessary. If you have a SDP on GitHub use the same email id while registering on Azure. If not complete the registration procedure with whichever option provided there. If you do use your institution mail id it may take a few days for Azure to verify. Having set that up , you should be redirected to the following screen.



Then using the hamburger menu on the left top corner navigate to Home which should bring you to the next screen which looks like the following .



At the navigate tab select Resource Groups , and then find the Add button towards the left corner to create a new resource group. There name your resource group what you like and select the region East US. The subscription plan is already selected for you in its respective box. Click on review and create and then if all validations are passed you can click on create and your resource group will be set up.

Next go back to Home and click on Machine Learning. Here you will find a similar screen to that of Resource Group. Again click on add and the following screen should be presented to you

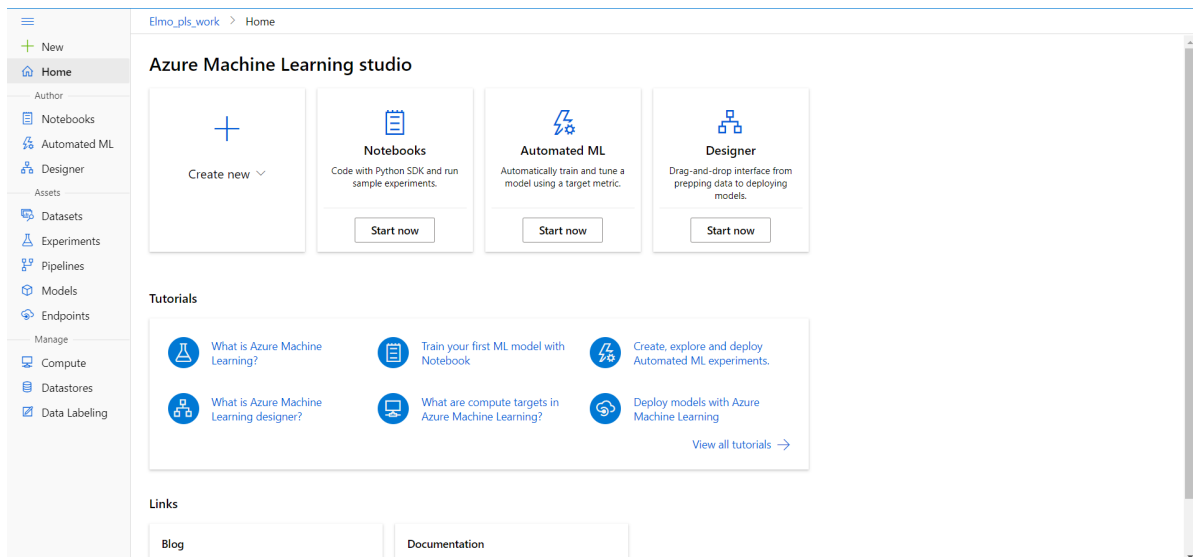
The screenshot shows the 'Machine Learning' workspace creation page in the Microsoft Azure portal. The page is titled 'Machine Learning' and 'Create a machine learning workspace'. It has two main sections: 'Project details' and 'Workspace details'. In the 'Project details' section, the 'Subscription' is set to 'Azure for Students' and the 'Resource group' is set to 'ELMO_group'. In the 'Workspace details' section, the 'Workspace name' is empty, with error messages indicating it is required and must be between 3 and 33 characters long, starting with an alphanumeric character. The 'Region' is set to 'East US'. The 'Storage account' is empty, with an error message indicating it must not be empty. The 'Key vault' is also empty. At the bottom, there are buttons for 'Review + create', '< Previous', and 'Next: Networking'.

Choose the resource group as the one you created and then name the Machine Learning project. Choose the region again as East US and rest of the fields will be filled in automatically. Then click on Review+Create and then create after validations are passed.

After this a loading screen will appear saying that your resource is being deployed. After it is deployed the system will notify you and on the notification box click on Take me to Resource. That should land you on a screen like this.

The screenshot shows the 'Overview' page for the 'elmopls' Machine Learning resource in the Microsoft Azure portal. The page has a left-hand navigation menu with options like 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Events', 'Settings', 'Private endpoint connections', 'Properties', 'Locks', 'Monitoring', 'Alerts', 'Metrics', 'Diagnostic settings', 'Logs', 'Automation', and 'Tasks (preview)'. The main content area is titled 'Essentials' and displays various properties of the resource, including 'Workspace edition', 'Resource group', 'Location', 'Subscription', 'Studio web URL', 'Storage', 'Registry', 'Key Vault', and 'Application Insights'. A 'Launch studio' button is prominently displayed at the bottom of the main content area.

Click on the Launch Studio button below and that will take you to the main page. This page should look like the following.



Click on Create New dropdown list and click on Compute Instance.

Create compute instance

Select virtual machine

centralindia

Virtual machine type ☐ CPU ☒ GPU

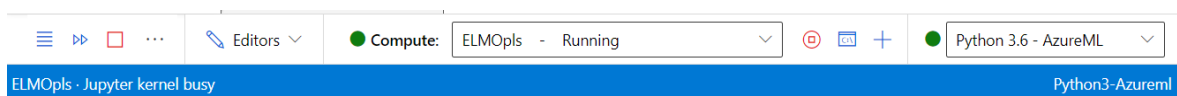
Virtual machine size ☐ Select from recommended options ☒ Select from all options

Showing 10 VM sizes Total available quota: 24 cores

| Name | GPU device | Available quota | Cost |
|--|-----------------------|-----------------|------------|
| You do not have enough quota for the following VM sizes. Click here to view and request quota. | | | |
| Standard_NC12s_v3 12 cores, 224GB RAM, 672GB storage | 2 x NVIDIA Tesla V100 | 0 cores | \$8.47/hr |
| Standard_NC24rs_v3 24 cores, 448GB RAM, 1344GB storage | 4 x NVIDIA Tesla V100 | 0 cores | \$18.63/hr |

Choose whichever machine you want to choose. Although in the Student pack only one GPU machine is allowed. After this click on next and then set the name for the machine and click on Create. It should take a few minutes to create the machine . After it is done you should see a Green Play button alongside the phrase Running appear along with your GPU name on the ML Studio Home. This means your VM is connected and running. Now you can start working on a notebook.

Click on Start Now on the Notebooks section and this will take you to the notebooks screen. Click on create new , name the file and choose the file type. Once your notebook is running , you will see that your notebook has connected to the Compute you had created and it is running. It generally connects to the Python3.6-Azure ML kernel and this can be viewed on the Kernel Screen.



With this your notebook is up and running, and you are ready to start your training for ELMO. Before that get your data ready for training. It is advisable to keep your training and validation sentences in separate CSV and then upload these CSV files onto your Azure Notebook Workspace.

Make sure you go through the BIL

[BILM-TF] (<https://github.com/allenai/bilm-tf>)

M-TF repository to know how data should be arranged. Here is another useful

[link] (<https://appliedmachinelearning.blog/2019/11/30/training-elmo-from-scratch-on-custom-data-set-for-generating-embeddings-tensorflow/>)

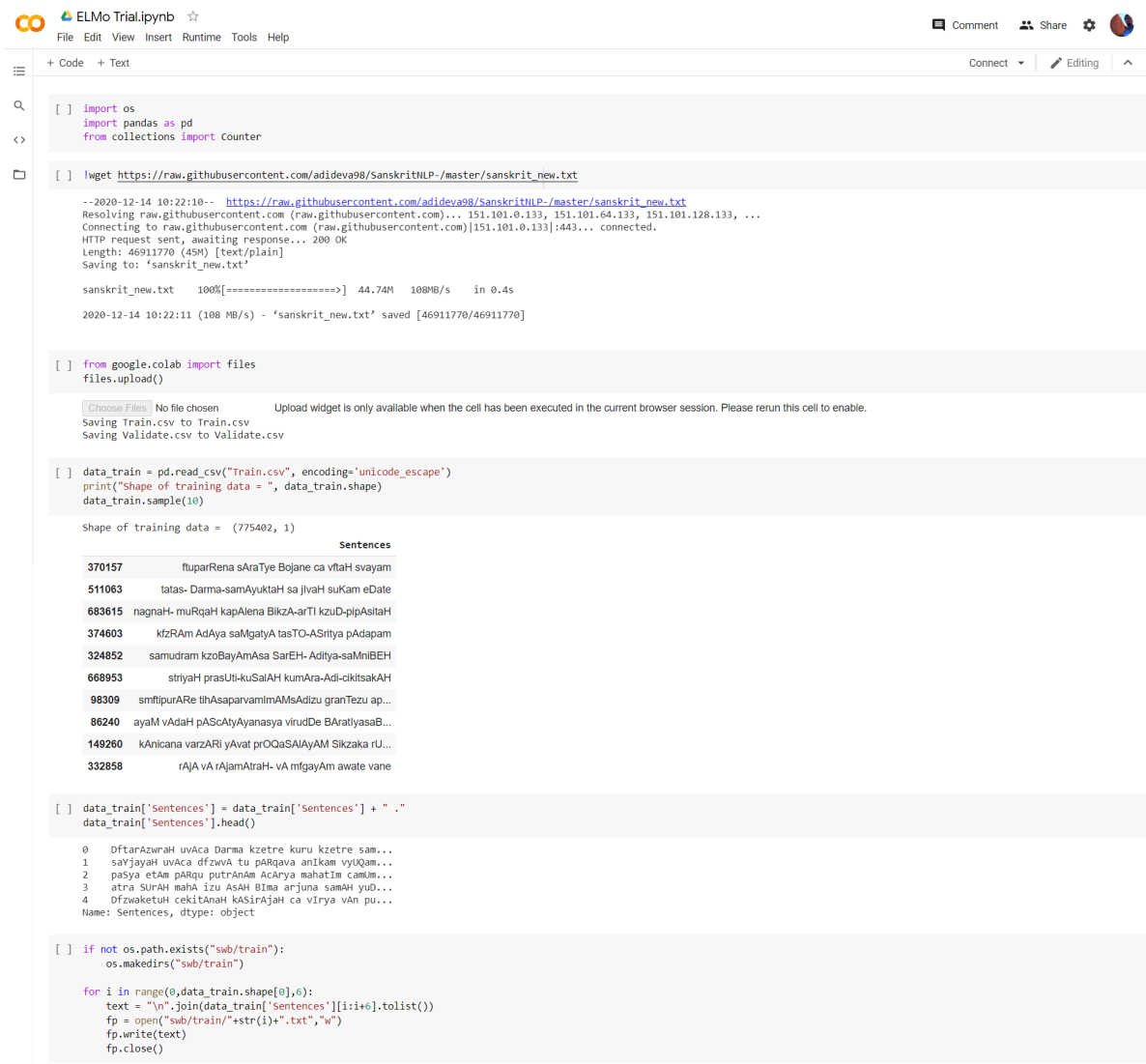
that guides on how to use the ELMO model. Follow the same steps verbatim to create a directory with training and validation holdout files, and the vocabulary file in the txt format. Follow the preprocessing steps provided in either links. The training directory creation on Azure should take between 5000-6000 (also depends on the VM and the tier of subscription)

With this your data is ready and now you can follow the steps in the BILM-TF. Remember to clone the repository before running any instructions.

Install python version 3.5 or later, tensorflow version 1.2 and h5py:

```
pip install tensorflow-gpu==1.2 h5py
python setup.py install
```

At this step remember to install TensorFlow 1.2 also , along with these. Also change directories as you work with these commands. Then run the next steps. Before running the training cell , ensure to specify the correct paths to your files. Depending on the machines you use and the size of your dataset it should take a while to train. As a standard, ELMO can train on 1 Billion words in 24 hours when using two NVIDIA P100 GPUs. A Collab notebook has been linked here to understand better how to implement the model.



The screenshot shows a Google Colab notebook titled "ELMo Trial.ipynb". The notebook is open to a cell containing the following code:

```
[ ] import os
import pandas as pd
from collections import Counter

[ ] wget https://raw.githubusercontent.com/adideva98/SanskritNLP-/master/sanskrit_new.txt

--2020-12-14 10:22:10-- https://raw.githubusercontent.com/adideva98/SanskritNLP-/master/sanskrit_new.txt
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.0.133, 151.101.64.133, 151.101.128.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)[151.101.0.133]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46911770 (45M) [text/plain]
Saving to: 'sanskrit_new.txt'

sanskrit_new.txt 100%[=====] 44.74M 108MB/s in 0.4s

2020-12-14 10:22:11 (108 MB/s) - 'sanskrit_new.txt' saved [46911770/46911770]

[ ] from google.colab import files
files.upload()

[ ] data_train = pd.read_csv("Train.csv", encoding='unicode_escape')
print("Shape of training data = ", data_train.shape)
data_train.sample(10)

Shape of training data = (775402, 1)

Sentences
370157 ftuparRena sArATye Bojane ca vItaH svayam
511063 tatas- Dama-samAyuktaH sa jIvaH suKam eDate
683615 nagnaH- muRqaH kapAlena BlkzA-arTI kzuD-plpAsItaH
374603 ktzRAM AdAya saMgatyA tasTO-ASritya pAdapam
324852 samudram kzoBayAmAsa SarEH- Aditya-saMnIBEH
668953 striyaH prasUtI-kuSAIAH kumAra-Adi-cikitsaKAH
98309 smItipurARe tItAsaparvamImAMsAdizu granTezu ap...
86240 ayaM vAdah pAScAtyAyanasya virudDe BAraItIyasaB...
149260 kAnicana varzARi yAvat prOQaSAIAyAM Sikzaka rU...
332858 rAJA vA rAJamAtraH- vA mIgayAm awate vane

[ ] data_train['Sentences'] = data_train['Sentences'] + " ."
data_train['Sentences'].head()

0 DftarAzwaH uvAca Dama kzetre kuru kzetre sam...
1 saYyajaH uvAca dfzwvA tu pARqava anIkam vyUQam...
2 paSyA eTAm pARqu putrAnAm ACArYA mahatIm camUm...
3 atra SURAH maha izu AsAH BIma arjuna samAH yud...
4 DfzwaKetuh cekItanAH kaSIArAJAH ca vIrya vAn pu...
Name: Sentences, dtype: object

[ ] if not os.path.exists("swb/train"):
os.makedirs("swb/train")

for i in range(0,data_train.shape[0],6):
text = "\n".join(data_train['Sentences'][i:i+6].tolist())
fp = open("swb/train/"+str(i)+".txt","w")
fp.write(text)
fp.close()
```

```
[ ] texts = " ".join(data_train['Sentences'].tolist())
words = texts.split(" ")
print("Number of tokens in Training data = ",len(words))
dictionary = Counter(words)
print("Size of Vocab",len(dictionary))
sorted_vocab = ["<S>","</S>","<UNK>"]
sorted_vocab.extend([pair[0] for pair in dictionary.most_common()])

text = "\n".join(sorted_vocab)
fp = open("swb/vocab.txt","w")
fp.write(text)
fp.close()

Number of tokens in Training data = 5724382
Size of Vocab 1082162

[ ] data_dev = pd.read_csv("Validate.csv")
data_dev['Sentences'] = data_dev['Sentences'] + " ."
if not os.path.exists("swb/dev"):
    os.makedirs("swb/dev")

for i in range(0,data_dev.shape[0],6):
    text = "\n".join(data_dev['Sentences'][i:i+6].tolist())
    fp = open("swb/dev/"+str(i)+".txt","w")
    fp.write(text)
    fp.close()

[ ] !git clone https://github.com/allenai/bilm-tf.git

Cloning into 'bilm-tf'...
remote: Enumerating objects: 292, done.
remote: Total 292 (delta 0), reused 0 (delta 0), pack-reused 292
Receiving objects: 100% (292/292), 588.40 KiB | 29.42 MiB/s, done.
Resolving deltas: 100% (137/137), done.

[ ] !pip install tensorflow-gpu==1.2
!pip install tensorflow==1.2
!pip install h5py

Collecting tensorflow-gpu==1.2
  Downloading https://files.pythonhosted.org/packages/cd/4d/c3c4da41c6d7b9a4949cb9e53c7032d7d9b7da0410f1226f7455209dd962/tensorflow-gpu-1.2.0-cp36-cp36m-manylinux1_x86_64.whl (89.5MB)
    | 89.5MB 33kB/s
Collecting markdown==2.2.0
  Downloading https://files.pythonhosted.org/packages/ac/99/288a81a38526a42c98b5b9832c6e339c8ad5dd38b19a53abfac7c8037c7f/Markdown-2.2.0.tar.gz (236kB)
    | 245kB 52.4kB/s
Requirement already satisfied: numpy>=1.11.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.2) (1.19.4)
Collecting bleach==1.5.0
  Downloading https://files.pythonhosted.org/packages/33/70/86c5fec937ea4964184d4d6c4f0b9551564f821e1c3575907639036d9b90/bleach-1.5.0-py2.py3-none-any.whl
Collecting backports.weakref==1.0rc1
  Downloading https://files.pythonhosted.org/packages/6a/f7/ae34b6818b603e264f26fe7db2bd07850ce331ce2fde74b266d61fa2d87/backports.weakref-1.0rc1-py3-none-any.whl
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.2) (1.15.0)
Requirement already satisfied: protobuf>=3.2.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.2) (3.12.4)
Requirement already satisfied: werkzeug>=0.11.10 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.2) (1.0.1)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.6/dist-packages (from tensorflow-gpu==1.2) (0.36.2)
Requirement already satisfied: html5lib==0.999999
  Downloading https://files.pythonhosted.org/packages/ae/ae/bcb60402c60932b32dfaf19bb53870b29eda2cd17551ba5639219fb5ebf9/html5lib-0.999999.tar.gz (889kB)
    | 889kB 52.4kB/s
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packages (from protobuf>=3.2.0->tensorflow-gpu==1.2) (51.0.0)
Building wheels for collected packages: markdown, html5lib
  Building wheel for markdown (setup.py) ... done
  Created wheel for markdown: filename=markdown-2.2.0-cp36-none-any.whl size=136295 sha256=da260b677d747af68f5522c0965c68d90ae30671b7b03b6126560bc6477144
  Stored in directory: /root/.cache/pip/wheels/b6/52/17/f0af18e3bec0cfaf60b361ffed15b4c3468f3b3c0b87fbc079
  Building wheel for html5lib (setup.py) ... done
  Created wheel for html5lib: filename=html5lib-0.999999-cp36-none-any.whl size=107222 sha256=a58183398518ec26acd8ec2a7c275877e755cf59b6de49bb70d6374f1cdf43e
  Stored in directory: /root/.cache/pip/wheels/50/ae/f9/d2b189788efcf61de0e36045476735c838898eef1cad6e29
Successfully built markdown html5lib
ERROR: tensorflow 2.4.0 has requirement markdown>=2.6.8, but you'll have markdown 2.2.0 which is incompatible.
Installing collected packages: markdown, html5lib, bleach, backports.weakref, tensorflow-gpu
  Found existing installation: Markdown-3.3.3
    Uninstalling Markdown-3.3.3:
      Successfully uninstalled Markdown-3.3.3
  Found existing installation: html5lib 1.0.1
    Uninstalling html5lib-1.0.1:
      Successfully uninstalled html5lib-1.0.1
  Found existing installation: bleach 3.2.1
    Uninstalling bleach-3.2.1:
      Successfully uninstalled bleach-3.2.1
Successfully installed backports.weakref-1.0rc1 bleach-1.5.0 html5lib-0.999999 markdown-2.2.0 tensorflow-gpu-1.2.0
Collecting tensorflow==1.2
  Downloading https://files.pythonhosted.org/packages/55/7995cc1e960fa37ea90e6777d832e75026fde5c6109215d892aaff2e9b7/tensorflow-1.2.0-cp36-cp36m-manylinux1_x86_64.whl (35.0MB)
    | 35.0MB 95kB/s
Requirement already satisfied: backports.weakref==1.0rc1 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.2) (1.0rc1)
Requirement already satisfied: protobuf>=3.2.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.2) (3.12.4)
Requirement already satisfied: werkzeug>=0.11.10 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.2) (1.0.1)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.2) (0.36.2)
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.2) (1.15.0)
Requirement already satisfied: numpy>=1.11.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.2) (1.19.4)
Requirement already satisfied: markdown==2.2.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.2) (2.2.0)
Requirement already satisfied: html5lib==0.999999 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.2) (0.999999)
Requirement already satisfied: bleach==1.5.0 in /usr/local/lib/python3.6/dist-packages (from tensorflow==1.2) (1.5.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-packages (from protobuf>=3.2.0->tensorflow==1.2) (51.0.0)
Installing collected packages: tensorflow
  Found existing installation: tensorflow 2.4.0
    Uninstalling tensorflow-2.4.0:
      Successfully uninstalled tensorflow-2.4.0
Successfully installed tensorflow-1.2.0
Requirement already satisfied: h5py in /usr/local/lib/python3.6/dist-packages (2.10.0)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from h5py) (1.15.0)
Requirement already satisfied: numpy>=1.7 in /usr/local/lib/python3.6/dist-packages (from h5py) (1.19.4)

[ ] cd bilm-tf
/content/bilm-tf

[ ] !python setup.py install

running install
running bdist_egg
running egg_info
creating bilm.egg-info
writing bilm.egg-info/PKG-INFO
writing dependency links to bilm.egg-info/dependency_links.txt
writing requirements to bilm.egg-info/requirements.txt
writing top-level names to bilm.egg-info/top_level.txt
writing manifest file 'bilm.egg-info/SOURCES.txt'
writing manifest file 'bilm.egg-info/SOURCES.txt'
installing library code to build/bdist.linux-x86_64/egg
running install_lib
running build_py
creating build
creating build/lib
creating build/lib/bilm
copying bilm/training.py -> build/lib/bilm
copying bilm/_init_.py -> build/lib/bilm
copying bilm/model.py -> build/lib/bilm
copying bilm/data.py -> build/lib/bilm
copying bilm/elmo.py -> build/lib/bilm
creating build/bdist.linux-x86_64
creating build/bdist.linux-x86_64/egg
creating build/bdist.linux-x86_64/egg/bilm
copying build/lib/bilm/training.py -> build/bdist.linux-x86_64/egg/bilm
copying build/lib/bilm/_init_.py -> build/bdist.linux-x86_64/egg/bilm
copying build/lib/bilm/model.py -> build/bdist.linux-x86_64/egg/bilm
copying build/lib/bilm/data.py -> build/bdist.linux-x86_64/egg/bilm
copying build/lib/bilm/elmo.py -> build/bdist.linux-x86_64/egg/bilm
byte-compiling build/bdist.linux-x86_64/egg/bilm/training.py to training.cpython-36.pyc
byte-compiling build/bdist.linux-x86_64/egg/bilm/_init_.py to _init_.cpython-36.pyc
byte-compiling build/bdist.linux-x86_64/egg/bilm/model.py to model.cpython-36.pyc
byte-compiling build/bdist.linux-x86_64/egg/bilm/data.py to data.cpython-36.pyc
byte-compiling build/bdist.linux-x86_64/egg/bilm/elmo.py to elmo.cpython-36.pyc
creating build/bdist.linux-x86_64/egg/EGG-INFO
copying bilm.egg-info/PKG-INFO -> build/bdist.linux-x86_64/egg/EGG-INFO
copying bilm.egg-info/SOURCES.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
copying bilm.egg-info/dependency_links.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
copying bilm.egg-info/not-zip-safe -> build/bdist.linux-x86_64/egg/EGG-INFO
copying bilm.egg-info/requirements.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
copying bilm.egg-info/top_level.txt -> build/bdist.linux-x86_64/egg/EGG-INFO
creating dist
creating 'dist/bilm-0.1.post5-py3.6.egg' and adding 'build/bdist.linux-x86_64/egg' to it
removing 'build/bdist.linux-x86_64/egg' (and everything under it)
Processing bilm-0.1.post5-py3.6.egg
creating /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg
```

```
Extracting bilm-0.1.post5-py3.6.egg to /usr/local/lib/python3.6/dist-packages
Adding bilm 0.1.post5 to easy-install.pth file

Installed /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg
Processing dependencies for bilm==0.1.post5
Searching for h5py==2.10.0
Best match: h5py 2.10.0
Adding h5py 2.10.0 to easy-install.pth file

Using /usr/local/lib/python3.6/dist-packages
Searching for six==1.15.0
Best match: six 1.15.0
Adding six 1.15.0 to easy-install.pth file
```

```
[ ] !python -m unittest discover tests/
```

```
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/dtypes.py:458: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of TensorFlow v2.0 this warning will become an error.
  _np_qint8 = np.dtype [("qint8", np.int8, 1)]
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/dtypes.py:459: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of TensorFlow v2.0 this warning will become an error.
  _np_quint8 = np.dtype [("quint8", np.uint8, 1)]
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/dtypes.py:460: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of TensorFlow v2.0 this warning will become an error.
  _np_qint16 = np.dtype [("qint16", np.int16, 1)]
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/dtypes.py:461: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of TensorFlow v2.0 this warning will become an error.
  _np_quint16 = np.dtype [("quint16", np.uint16, 1)]
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/dtypes.py:462: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of TensorFlow v2.0 this warning will become an error.
  _np_qint32 = np.dtype [("qint32", np.int32, 1)]
/usr/local/lib/python3.6/dist-packages/tensorflow/python/framework/dtypes.py:465: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of TensorFlow v2.0 this warning will become an error.
  _np_resource = np.dtype [("resource", np.ubyte, 1)]
Found 1 shards at /tmp/tmpklpr7wqf
Loading data from: /tmp/tmpklpr7wqf
Loaded 3 sentences.
Finished loading
Found 1 shards at /tmp/tmpklpr7wqf
Loading data from: /tmp/tmpklpr7wqf
Loaded 3 sentences.
Finished loading
Loading data from: /tmp/tmpklpr7wqf
Loaded 3 sentences.
Finished loading
Loading data from: /tmp/tmpklpr7wqf
Loaded 3 sentences.
Finished loading
Found 1 shards at /tmp/tmpklpr7wqf
Loading data from: /tmp/tmpklpr7wqf
Loaded 3 sentences.
Finished loading
Loading data from: /tmp/tmpklpr7wqf
Loaded 3 sentences.
Finished loading
Found 1 shards at /tmp/tmpklpr7wqf
Loading data from: /tmp/tmpklpr7wqf
Loaded 3 sentences.
Finished loading
Loading data from: /tmp/tmpklpr7wqf
Loaded 3 sentences.
Finished loading
Found 1 shards at /tmp/tmpklpr7wqf
Loading data from: /tmp/tmpklpr7wqf
Loaded 3 sentences.
Finished loading
Loading data from: /tmp/tmpklpr7wqf
Loaded 3 sentences.
Finished loading
Found 1 shards at /tmp/tmpklpr7wqf
Loading data from: /tmp/tmpklpr7wqf
Loaded 3 sentences.
Finished loading
Found 1 shards at /tmp/tmpklpr7wqf
```

```
[ ] %tensorflow_version 1.2
import tensorflow as tf
device_name = tf.test.gpu_device_name()
if device_name != '/device:GPU:0':
    raise SystemError('GPU device not found')
print('Found GPU at: {}'.format(device_name))

%tensorflow_version only switches the major version: 1.x or 2.x.
You set: '1.2'. This will be interpreted as: '1.x'.
```

```
TensorFlow 1.x selected.
Found GPU at: /device:GPU:0
```

```
[ ] from tensorflow.python.client import device_lib
device_lib.list_local_devices()

[{"name": "/device:CPU:0",
  device_type: "CPU",
  memory_limit: 268435456
  locality {
  }
  incarnation: 12891412254140096734, name: "/device:XLA_CPU:0"
  device_type: "XLA_CPU",
  memory_limit: 17179869184
  locality {
  }
  incarnation: 4978776110176278321
  physical_device_desc: "device: XLA_CPU device", name: "/device:XLA_GPU:0"
  device_type: "XLA_GPU",
  memory_limit: 17179869184
  locality {
  }
  incarnation: 9439870788999271479
  physical_device_desc: "device: XLA_GPU device", name: "/device:GPU:0"
  device_type: "GPU",
  memory_limit: 14912199066
  locality {
    bus_id: 1
    links {
    }
  }
  incarnation: 2699072999230041613
  physical_device_desc: "device: 0, name: Tesla T4, pci bus id: 0000:00:04:0, compute capability: 7.5"]
```

```
[ ] cd /content/bilm-tf/bin
```

```
/content/bilm-tf/bin
```

```
[ ]
```

```
!python /content/bilm-tf/bin/train_elmo.py --train_prefix='/content/swb/train/*' --vocab_file '/content/swb/vocab.txt' --save_dir '/content/swb/checkpoint'
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:21: The name tf.logging.set_verbosity is deprecated. Please use tf.compat.v1.logging.set_verbosity instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:21: The name tf.logging.INFO is deprecated. Please use tf.compat.v1.logging.INFO instead.

Found 129234 shards at /content/swb/train/*
Loading data from: /content/swb/train/155976.txt
Loaded 6 sentences.
Finished loading
Found 129234 shards at /content/swb/train/*
Loading data from: /content/swb/train/274776.txt
Loaded 6 sentences.
Finished loading
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:684: The name tf.get_variable is deprecated. Please use tf.compat.v1.get_variable instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:690: The name tf.train.AdagradOptimizer is deprecated. Please use tf.compat.v1.train.AdagradOptimizer instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:702: The name tf.variable_scope is deprecated. Please use tf.compat.v1.variable_scope instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:153: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:224: The name tf.get_variable_scope is deprecated. Please use tf.compat.v1.get_variable_scope instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:224: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool_v2 instead.
```

```

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:411: the name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/python/util/dispatch.py:180: calling squeeze (from tensorflow.python.ops.array_ops) with squeeze_dims is deprecated
Instructions for updating:
Use the 'axis' argument instead
USING SKIP CONNECTIONS
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:372: LSTMCell.__init__ (from tensorflow.python.ops.rnn_cell_impl) is deprecated
Instructions for updating:
This class is equivalent as tf.keras.layers.LSTMCell, and will be replaced by that in Tensorflow 2.0.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:390: The name tf.nn.rnn_cell.DropoutWrapper is deprecated. Please use tf.compat.v1.nn.rnn_cell.DropoutWrapper instead.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:386: The name tf.nn.rnn_cell.ResidualWrapper is deprecated. Please use tf.compat.v1.nn.rnn_cell.ResidualWrapper instead.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:396: MultiRNNCell.__init__ (from tensorflow.python.ops.rnn_cell_impl) is deprecated
Instructions for updating:
This class is equivalent as tf.keras.layers.StackedRNNCells, and will be replaced by that in Tensorflow 2.0.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:410: static_rnn (from tensorflow.python.ops.rnn) is deprecated and will be removed in a future version
Instructions for updating:
Please use 'keras.layers.RNN(cell, unroll=True)', which is equivalent to this API
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/python/ops/rnn_cell_impl.py:958: Layer.add_variable (from tensorflow.python.keras.engine.base_layer) is deprecated
Instructions for updating:
Please use 'layer.add_weight' method instead.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/python/ops/rnn_cell_impl.py:962: calling Zeros.__init__ (from tensorflow.python.ops.init_ops) with dtype is deprecated
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:424: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated
Instructions for updating:
Please use 'rate' instead of 'keep_prob'. Rate should be set to 'rate = 1 - keep_prob'.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:425: The name tf.add_to_collection is deprecated. Please use tf.compat.v1.add_to_collection instead.
WARNING:tensorflow:From /tensorflow-1.15.2/python3.6/tensorflow_core/python/ops/math_grad.py:1424: where (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:Variable += will be deprecated. Use variable.assign_add if you want assignment to the variable value or 'x = x + y' if you want a new python Tensor object.
USING SKIP CONNECTIONS
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/bilm-0.1.post5-py3.6.egg/bilm/training.py:26: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

```

The TensorFlow magic words can be used only on Colab, however the device listing command can be used on any Jupyter Notebook. The Colab Notebook can be found

[here] (<https://colab.research.google.com/drive/1SNEqzdw21pYMI3XaQJIEtYgYHmK6bvHY#scrollTo=rkJukjoWOuas>)