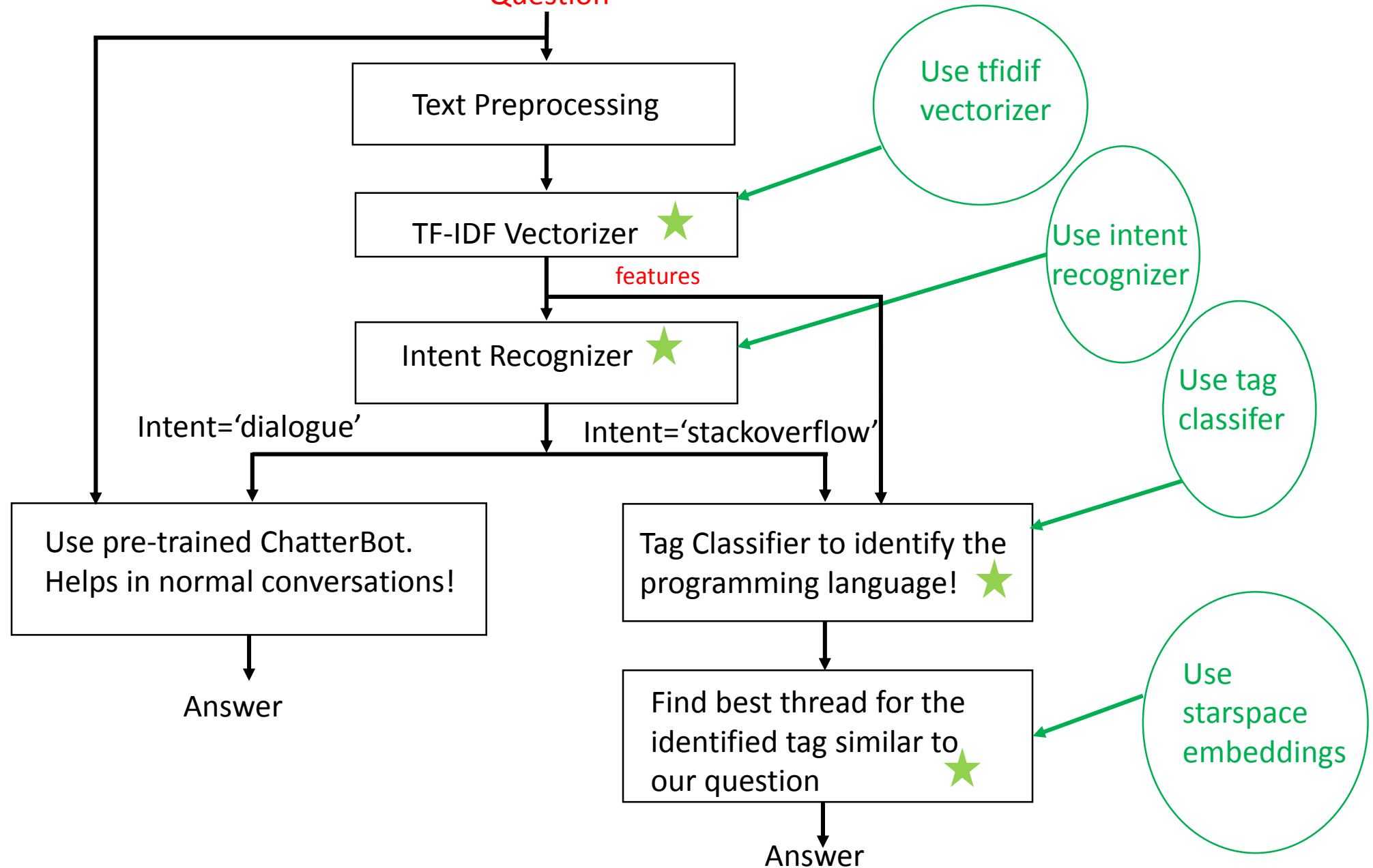


How does the stackoverflow bot assistant function?



- To understand the above flow of chatbot working we need to understand various building blocks and how they are prepared to work!
- First things first, this chat-bot project is set-up on Telegram and AWS.
- **Telegram:**
 - Integrating the bot to telegram requires us to generate a token.
 - First we need to create a personal account on Telegram and then talk to @BotFather
 - The command "/newbot" will create a bot for you. You will be prompted to enter a name and a username for your bot. After that, you will be given a token.
 - When you will have *main_bot.py* script ready you can run

```
python3 main_bot.py --token=YOUR_TOKEN
```

to start interacting with bot.

BotFather

bot

December 20, 2018

/start 2:50 PM ✓✓

I can help you create and manage Telegram bots. If you're new to the Bot API, please [see the manual](#).

You can control me by sending these commands:

[/newbot](#) - create a new bot
[/mybots](#) - edit your bots [beta]

Edit Bots

[/setname](#) - change a bot's name
[/setdescription](#) - change bot description
[/setabouttext](#) - change bot about info
[/setuserpic](#) - change bot profile photo
[/setcommands](#) - change the list of commands
[/deletebot](#) - delete a bot

Bot Settings

[/token](#) - generate authorization token
[/revoke](#) - revoke bot access token
[/setinline](#) - toggle inline mode
[/setinlinegeo](#) - toggle inline location requests
[/setinlinefeedback](#) - change inline feedback settings
[/setjoininggroups](#) - can your bot be added to groups?
[/setprivacy](#) - toggle privacy mode in groups

Games

[/mygames](#) - edit your games [beta]
[/newgame](#) - create a new game
[/listgames](#) - get a list of your games
[/editgame](#) - edit a game
[/deletgame](#) - delete an existing game

2:50 PM

/newbot 2:59 PM ✓✓

BotFather

bot

Alright, a new bot. How are we going to call it? Please choose a name for your bot.

2:59 PM

Astronaut Vini 3:00 PM ✓✓

Good. Now let's choose a username for your bot. It must end in `bot`. Like this, for example: TetrisBot or tetris_bot.

3:00 PM

vini1522_bot 3:01 PM ✓✓

Done! Congratulations on your new bot. You will find it at t.me/vini1522_bot. You can now add a description, about section and profile picture for your bot, see [/help](#) for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:

791620559:AAGqIQk62I_zVqY5rsjPHSjqhJkZowu5dts

For a description of the Bot API, see this page:

<https://core.telegram.org/bots/api>

3:01 PM

/setdescription 3:22 PM ✓✓

Choose a bot to change description. 3:22 PM

@vini1522_bot 3:23 PM ✓✓

OK. Send me the new description for the bot. People will see this description when they open a chat with your bot, in a block titled 'What can this bot do?'.

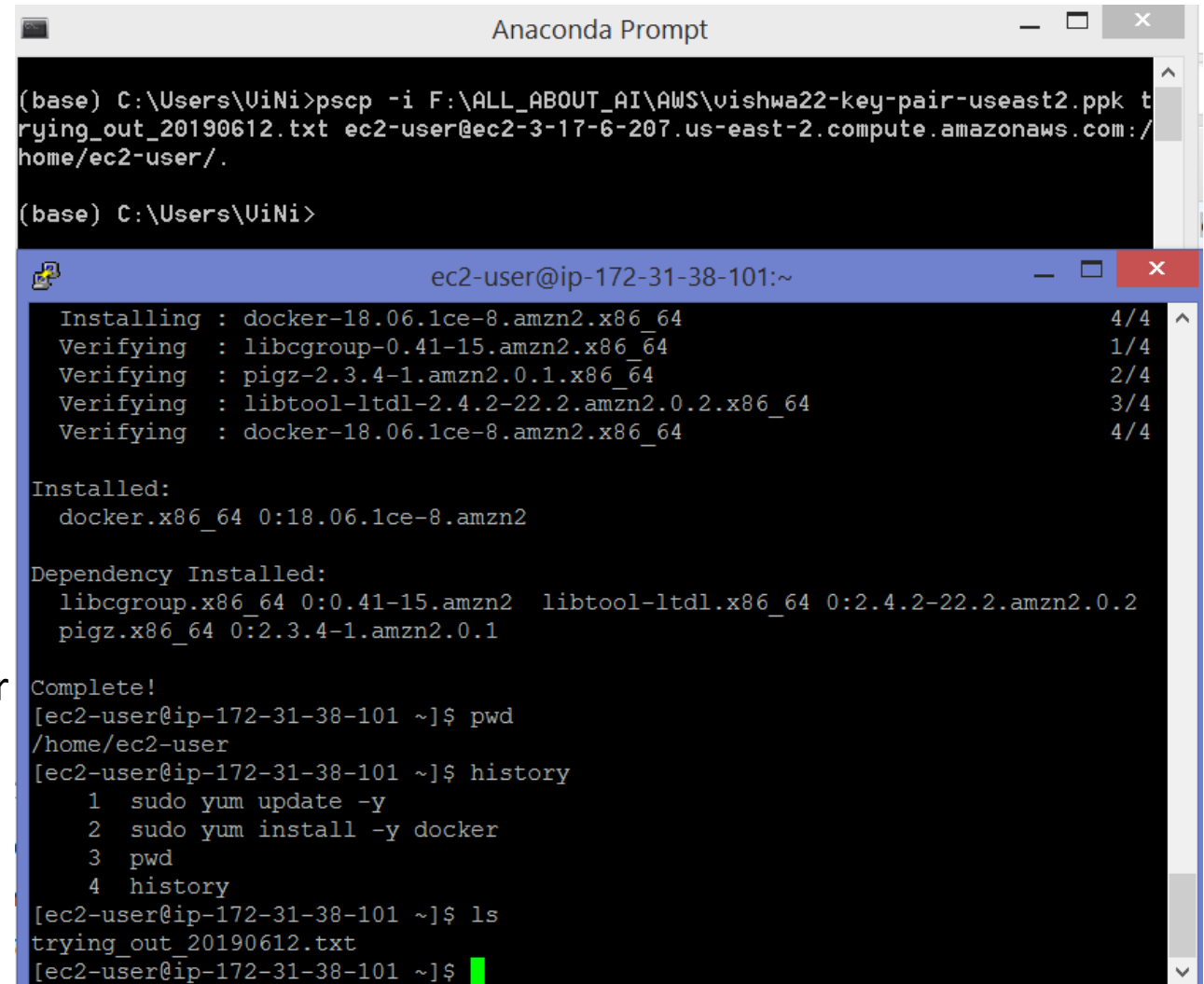
3:23 PM

This is my FIRST bot! 3:23 PM ✓✓

Success! Description updated. [/help](#) 3:23 PM

• AWS:

- We will be hosting our bot on AWS. We need to sign up at AWS console and launch a free tier instance (Ubuntu Server 16.04 LTS) using the instructions given here: <https://github.com/hse-aml/natural-language-processing/blob/master/AWS-tutorial.md>
- Here we are working on windows 8.1 and to interact with AWS Linux instance for transferring files we need a PuTTY (a free SSH client for Windows) using the instructions given here: <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html>
- In this figure we see example of transferring a file from our local Machine to AWS linux machine
- On AWS linux machine we can install docker and pull the docker image ***akashin/coursera-aml-nlp*** to set up with all the dependencies



The image shows two overlapping terminal windows. The top window is 'Anaconda Prompt' on a Windows machine, showing the command to use 'pscp' to transfer a private key file from a local drive to an AWS EC2 instance. The bottom window is a PuTTY terminal connected to an AWS Linux instance (ip-172-31-38-101). It displays the successful installation of Docker and its dependencies (libcgrouper, pigz, libtool-ltdl) on an Amazon Linux 2 system. The terminal output includes progress bars for each package, a list of installed packages, and a confirmation message 'Complete!'. Finally, the user runs 'ls' to verify the file transfer, showing 'trying_out_20190612.txt' in the current directory.

```
(base) C:\Users\UiNi>pscp -i F:\ALL_ABOUT_AI\AWS\vishwa22-key-pair-useast2.ppk t
rying_out_20190612.txt ec2-user@ec2-3-17-6-207.us-east-2.compute.amazonaws.com:/
home/ec2-user/.

(base) C:\Users\UiNi>

ec2-user@ip-172-31-38-101:~
Installing : docker-18.06.1ce-8.amzn2.x86_64 4/4
Verifying : libcgrouper-0.41-15.amzn2.x86_64 1/4
Verifying : pigz-2.3.4-1.amzn2.0.1.x86_64 2/4
Verifying : libtool-ltdl-2.4.2-22.2.amzn2.0.2.x86_64 3/4
Verifying : docker-18.06.1ce-8.amzn2.x86_64 4/4

Installed:
  docker.x86_64 0:18.06.1ce-8.amzn2

Dependency Installed:
  libcgrouper.x86_64 0:0.41-15.amzn2  libtool-ltdl.x86_64 0:2.4.2-22.2.amzn2.0.2
  pigz.x86_64 0:2.3.4-1.amzn2.0.1

Complete!
[ec2-user@ip-172-31-38-101 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-38-101 ~]$ history
 1 sudo yum update -y
 2 sudo yum install -y docker
 3 pwd
 4 history
[ec2-user@ip-172-31-38-101 ~]$ ls
trying_out_20190612.txt
[ec2-user@ip-172-31-38-101 ~]$
```

- Setup docker image on AWS linux instance

```
[ec2-user@ip-172-31-38-101 ~]$ docker pull akashin/coursera-aml-nlp
Using default tag: latest
latest: Pulling from akashin/coursera-aml-nlp
aafe6b5e13de: Pull complete
0a2b43a72660: Pull complete
18bdd1e546d2: Pull complete
8198342c3e05: Pull complete
f56970a44fd4: Pull complete
0ebdb632688b: Pull complete
b19fb64301fe: Pull complete
09369b2a81ba: Pull complete
692ff49613b4: Pull complete
7b6a4c2ad4d9: Pull complete
0670dae5c2ff: Pull complete
8308536230cc: Pull complete
917b9fd049d8: Pull complete
e1e5ba2729ba: Pull complete
39adb459aefb: Pull complete
f935aae7a64c: Pull complete
bd6d120d9da8: Pull complete
b75aaa4e6564: Pull complete
ca48c9546bf5: Pull complete
abfab7b0ab7: Pull complete
Digest: sha256:d5b7f4c75a9786982fc658814d56512dbb1387853b563f8e330c99ef56f47cf0
Status: Downloaded newer image for akashin/coursera-aml-nlp:latest
[ec2-user@ip-172-31-38-101 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-38-101 ~]$ ls
trying_out_20190612.txt
[ec2-user@ip-172-31-38-101 ~]$ docker run -it -p 8080:8080 --name coursera-aml-nlp -v $PWD:/root/coursera akashin/coursera-aml-nlp

=====
Welcome to the Docker container for the Coursera NLP course.

This container contains dependencies that you might need
to complete course assignments.

You can also install any additional system dependencies with
> apt-get install PACKAGE_NAME

And Python dependencies with
> pip3 install PACKAGE_NAME

To run Jupyter Notebook in the container just type
> run_notebook
=====
root@c5b9ec263632:~#
```

- Once telegram bot and AWS linux is ready we can focus on the main goal i.e. to prepare models to run the bot.
- Dataset provided:
 - dialogues.tsv ➡ Dialogue phrases from movie subtitles (*negative samples*).
 - Tagged_posts.tsv ➡ StackOverflow posts, tagged with one programming language (*positive samples*).
- Checkout [*Chatbot_project.ipynb*](#). It performs specifically two tasks:
 - Intent and Language Recognition
 1. First we need to distinguish programming related questions from general ones.
 2. Later the programming related questions need to be tagged with the corresponding programming language (here only one!)
 - Ranking questions with embeddings
 1. To find a relevant answer (a thread from StackOverflow) on a question you will use vector representations to calculate similarity between the question and existing threads.
 2. To find similarity between texts means to find cosine similarity between their representative vectors. These vectors are Starspace embeddings trained specifically on Stack Overflow posts.

- As a result of [Chatbot_project.ipynb](#) we will obtain
 - Tfidf Vectorizer
 - Intent recognizer
 - Tag classifier
 - Folder containing embeddings of stackoverflow posts/threads segregated “tag” wise. This saves lot of time at runtime in generating the embeddings needed to compare various stackoverflow threads and rank them as per the closest similarity to our question.

c_cpp.pkl java.pkl php.pkl r.pkl swift.pkl
c#.pkl javascript.pkl python.pkl ruby.pkl vb.pkl

counts_by_tag

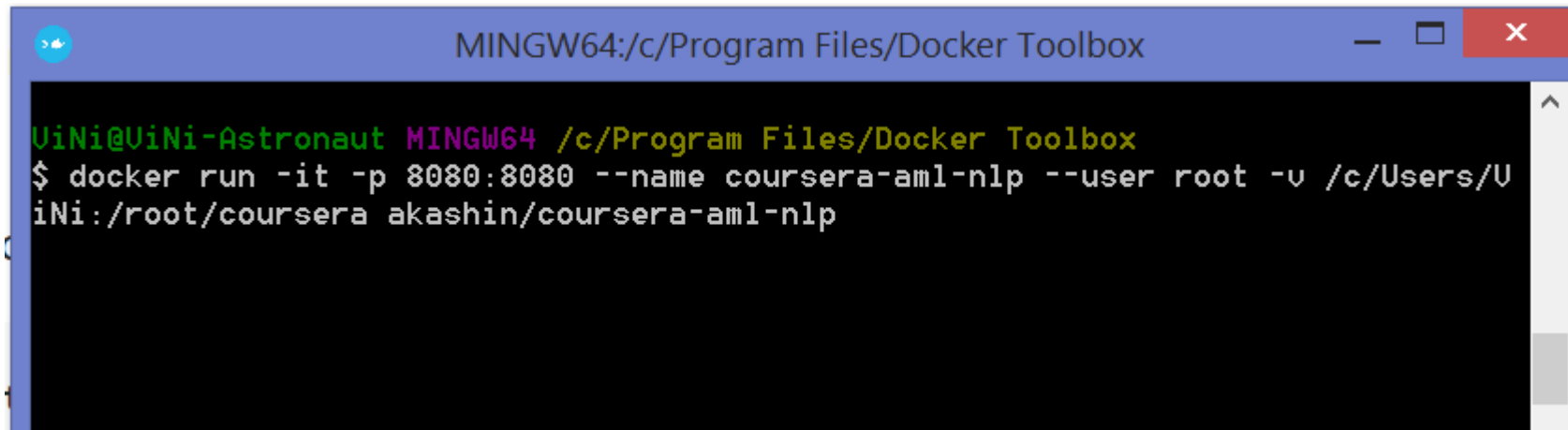
tag	
c#	394451
c_cpp	281300
java	383456
javascript	375867
php	321752
python	208607
r	36359
ruby	99930
swift	34809
vb	35044
Name: title, dtype: int64	

- Details of starspace embeddings, how they are generated and how they work is mentioned in the following notebook:

[Starspace_embeddings_Stackoverflow.ipynb](#)

- Basically we need StarSpace neural model (<https://github.com/facebookresearch/StarSpace>) to train embeddings on the given dataset (stackoverflow posts)
- Since Starspace cannot run on windows, we have a docker image *akashin/courser-aml-nlp* which contains Starspace and can help us generate embeddings file.
- Please read below the basic steps to follow in Docker Toolbox (in your local machine)

- Install docker toolbox for your windows system
- Pull the necessary docker image using the command:
docker pull akashin/coursera-aml-nlp
- Then run the docker file on your windows users directory
/c/Users/<user-name>



```
UiNi@UiNi-Astronaut MINGW64 /c/Program Files/Docker Toolbox
$ docker run -it -p 8080:8080 --name coursera-aml-nlp --user root -v /c/Users/UiNi:/root/coursera akashin/coursera-aml-nlp
iNi:/root/coursera
```

- Make sure the training file is present in the <user-name> directory
- Once you enter the docker go to folder coursera using *cd coursera*

```
root@17b833a88e92:~/coursera# ls prepared_train.tsv
prepared_train.tsv
root@17b833a88e92:~/coursera# starspace -help
Here is the help! Usage:

"starspace train ..." or "starspace test ..."

The following arguments are mandatory for train:
  -trainFile      training file path
  -model          output model file path

The following arguments are mandatory for test:
  -testFile       test file path
  -model          model file path

The following arguments for the dictionary are optional:
  -minCount       minimal number of word occurrences [1]
  -minCountLabel  minimal number of label occurrences [1]
  -ngrams         max length of word ngram [1]
  -bucket         number of buckets [20000000]
  -label          labels prefix [__label__]
```

Run the starspace model with first argument as *train* and list out other necessary *parameters* as shown below:

```
root@17b833a88e92:~/coursera# starspace train -trainFile prepared_train.tsv -model my_starspace_embeddings_0612 -trainMode 3 -adagrad true -ngrams 1 -epoch 5 -dim 100 -similarity cosine -minCount 2 -verbose true -fileFormat labelDoc -negSearchLimit 10 -lr 0.05
```

Starspace model and embeddings

```
MINGW64:/c/Program Files/Docker Toolbox
Build dict from input file : prepared_train.tsv
Read 12M words
Number of words in dictionary: 95058
Number of labels in dictionary: 0
Loading data from file : prepared_train.tsv
Total number of examples loaded : 999740
Initialized model weights. Model size :
matrix : 95058 100
Training epoch 0: 0.05 0.01
Epoch: 100.0% lr: 0.040000 loss: 0.008951 eta: 0h48m tot: 0h12m9s (20.0%)
----- Epoch 0 Train error : 0.00894883 +++---
Training epoch 1: 0.04 0.01
Epoch: 100.0% lr: 0.030000 loss: 0.002623 eta: 0h33m tot: 0h23m18s (40.0%)
----- Epoch 1 Train error : 0.00265915 +++---
Training epoch 2: 0.03 0.01
Epoch: 100.0% lr: 0.020000 loss: 0.001918 eta: 0h20m tot: 0h33m46s (60.0%)
----- Epoch 2 Train error : 0.00189000 +++---
Training epoch 3: 0.02 0.01
Epoch: 100.0% lr: 0.010000 loss: 0.001578 eta: 0h10m tot: 0h44m16s (80.0%)
----- Epoch 3 Train error : 0.00155673 +++---
Training epoch 4: 0.01 0.01
Epoch: 100.0% lr: 0.000000 loss: 0.001371 eta: <1min tot: 0h55m19s (100.0%)
)
----- Epoch 4 Train error : 0.00137673 +++---
Saving model to file : my_starspace_embeddings_0612
Saving model in tsv format my_starspace_embeddings_0612.tsv
root@17b833a88e92:~/coursera#
```

Starspace embeddings
trained from scratch
specific to the
stackoverflow posts
data

Once all the necessary models are ready!

- Send them to aws linux machine. For example see below how we send `dialogue_manager.py` and `main_bot.py` to AWS from our local machine. Similar can be done for models and embeddings

```
(base) C:\Users\UiNi>pscp -i F:\ALL_ABOUT_AI\AWS\vishwa22-key-pair-useast2.ppk F:\HSE-AMLNLP\chatbot-new-2702\sendtoaws2\* ec2-user@ec2-3-17-6-207.us-east-2.compute.amazonaws.com:/home/ec2-user
dialogue_manager.py      | 3 kB | 3.9 kB/s | ETA: 00:00:00 | 100%
main_bot.py             | 3 kB | 3.6 kB/s | ETA: 00:00:00 | 100%
```

- Check them on AWS machine

```
root@c5b9ec263632:~# ls coursera/
dialogue_manager.py  tag_classifier.pkl      trying_out_20190612.txt
intent_recognizer.pkl  tfidf_vectorizer.pkl   utils.py
main_bot.py          thread_embeddings_by_tags  word_embeddings.tsv
root@c5b9ec263632:~# ls coursera/thread_embeddings_by_tags/
c#.pkl      java.pkl      php.pkl      r.pkl      swift.pkl
c_cpp.pkl   javascript.pkl  python.pkl  ruby.pkl   vb.pkl
root@c5b9ec263632:~#
```

Ready to talk to your bot!

```
root@c5b9ec263632:~# cd coursera
root@c5b9ec263632:~/coursera# python3 main_bot.py --token=791620559:AAGqIQk62I_zVqY5rsjPHSjqhJkZow
u5dts
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Loading resources...
/usr/local/lib/python3.5/dist-packages/sklearn/base.py:311: UserWarning: Trying to unpickle estimator
LogisticRegression from version 0.20.2 when using version 0.19.1. This might lead to breaking
code or invalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.5/dist-packages/sklearn/base.py:311: UserWarning: Trying to unpickle estimator
TfidfTransformer from version 0.20.2 when using version 0.19.1. This might lead to breaking co
de or invalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.5/dist-packages/sklearn/base.py:311: UserWarning: Trying to unpickle estimator
TfidfVectorizer from version 0.20.2 when using version 0.19.1. This might lead to breaking cod
e or invalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.5/dist-packages/sklearn/base.py:311: UserWarning: Trying to unpickle estimator
LabelBinarizer from version 0.20.2 when using version 0.19.1. This might lead to breaking code
or invalid results. Use at your own risk.
  UserWarning)
/usr/local/lib/python3.5/dist-packages/sklearn/base.py:311: UserWarning: Trying to unpickle estimator
OneVsRestClassifier from version 0.20.2 when using version 0.19.1. This might lead to breaking
code or invalid results. Use at your own risk.
  UserWarning)
Ready to talk!
An update received.
```

June 12

/start 11:48 AM ✓✓

AV Hi, I am your project bot. How can I help you today? 11:48 AM

How are you? 11:50 AM ✓✓

AV Who? Who is but a form following the function of what 11:50 AM

How are you doing? 11:50 AM ✓✓

AV I am doing well. 11:51 AM

Can I ask you few questions? 11:51 AM ✓✓

AV Go ahead and ask. 11:51 AM

multithreading in Java 11:51 AM ✓✓

I think its about ['java']
This thread might help you:
<https://stackoverflow.com/questions/5692521>

Stack Overflow

multithreading in java

I want "runnable" to run at 5tps. This is not
executing paralelly.

...



11:51 AM



How to sort a dictionary using values 11:52 AM ✓✓

I think its about ['python']
This thread might help you:
<https://stackoverflow.com/questions/4331892>

Stack Overflow

Python: sort this dictionary (dict in dict)

```
d = { 'a': {'k': 1, 'b': 'whatever'}, 'b': {'k': 2, 'b': 'sort by k'} }
```

Want to sort this dictionary by k as descending ...



11:52 AM



Convert list into set 11:52 AM ✓✓

I think its about ['python']
This thread might help you:
<https://stackoverflow.com/questions/33125611>

Stack Overflow

How to convert list into set in pandas?

I have a dataframe as below:

date

uids...



11:52 AM

