# NLP

# Spacy installation

$ pip install spacy

* Run anaconda cmd in administrator mode

$ conda info –envs

* Choose environment (eg: C:\Users\abhijith.m\AppData\Local\conda\conda\envs\python36)
* Activate environment

$ activate C:\Users\abhijith.m\AppData\Local\conda\conda\envs\python36

* Output : (python36) C:\Windows\system32>

$ pip install <https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-2.0.0/en_core_web_sm-2.0.0.tar.gz#egg=en_core_web_sm>

or

$ python -m spacy download en\_core\_web\_sm

* and in your python file:
* import en\_core\_web\_sm
* nlp = en\_core\_web\_sm.load()

# Morphological analysis

## Polyglot \*\*

### Installation

$ pip install polyglot

* Download this files:
* <https://download.lfd.uci.edu/pythonlibs/g5apjq5m/PyICU-2.3.1-cp36-cp36m-win_amd64.whl>
* <https://download.lfd.uci.edu/pythonlibs/g5apjq5m/pycld2-0.31-cp36-cp36m-win_amd64.whl>
* If it is not working go to this site and download apropriate PyICU and pycld2 file
* <https://www.lfd.uci.edu/~gohlke/pythonlibs/>
* pip install pycld2-0.31-cp36-cp36m-win\_amd64.whl
* pip install Morfessor-2.0.4-py2.py3-none-any.whl
* git clone <https://github.com/aboSamoor/polyglot.git>
* cd polyglot
* python setup.py install

Or try using this method

* pip install polyglot

PyICU wraps the ICU (International Components for Unicode) library.

PyICU‑2.3.1‑cp27‑cp27m‑win32.whl

PyICU‑2.3.1‑cp27‑cp27m‑win\_amd64.whl

PyICU‑2.3.1‑cp35‑cp35m‑win32.whl

PyICU‑2.3.1‑cp35‑cp35m‑win\_amd64.whl

PyICU‑2.3.1‑cp36‑cp36m‑win32.whl

PyICU‑2.3.1‑cp36‑cp36m‑win\_amd64.whl

PyICU‑2.3.1‑cp37‑cp37m‑win32.whl

PyICU‑2.3.1‑cp37‑cp37m‑win\_amd64.whl

the 27 means Python 2.7 and the 36 Python 3.6... If you have 64 bits python and windows then choose the amd64 otherwhise the win32 version.

* polyglot download embeddings2.en
* polyglot download ner2.en

### Morphological Analysis

### Download Necessary Models

* polyglot download morph2.en

### Example

<https://polyglot.readthedocs.io/en/latest/MorphologicalAnalysis.html>

from polyglot.text import Text, Word

blob = "Wewillmeettoday."

text = Text(blob)

text.language = "en"

text.morphemes

* WordList([u'We', u'will', u'meet', u'to', u'day', u'.'])

# Named Entity Recognition \*\*\*\*\*

Using spaCy

<https://towardsdatascience.com/named-entity-recognition-with-nltk-and-spacy-8c4a7d88e7da>

# document similarity

## Gensim

### Prerequisites[¶](https://radimrehurek.com/gensim/simserver.html#prerequisites)

* pip install simserver

OR

* git clone <https://github.com/RaRe-Technologies/gensim-simserver.git>
* cd simserver

**>>> from** **simserver** **import** SessionServer

**>>>** server = SessionServer('/tmp/my\_server') *# resume server (or create a new one)*

It is assumed you have gensim properly [**installed**](https://radimrehurek.com/gensim/install.html). You’ll also need the **[sqlitedict](https://pypi.python.org/pypi/sqlitedict)** package that wraps Python’s sqlite3 module in a thread-safe manner:

* pip install sqlitedict

To test the remote server capabilities, install Pyro4 (Python Remote Objects, at version 4.8 as of this writing):

$ sudo easy\_install Pyro4

* pip install Pyro4

Don’t forget to initialize logging to see logging messages:

**>>> import** **logging**

**>>>** logging.basicConfig(format='*%(asctime)s* : *%(levelname)s* : *%(message)s*', level=logging.INFO)

# Logic Programming

## **Kanren**

* **Kanren-** It lets us express logic as rules and facts and simplifies making code for business logic.
* pip install kanren
* **SymPy-** This is a Python library for symbolic mathematics. It is nearly a full-featured Computer Algebra System.
* pip install sympy

# Chat bot

## Memory Network

<https://www.youtube.com/watch?v=BN7Kp0JD04o>

1. github code: <https://github.com/domluna/memn2n>

git clone git@github.com:domluna/memn2n.git

mkdir ./memn2n/data/

cd ./memn2n/data/

wget http://www.thespermwhale.com/jaseweston/babi/tasks\_1-20\_v1-2.tar.gz

tar xzvf ./tasks\_1-20\_v1-2.tar.gz

cd ../

python single.py

1. github code: <https://github.com/carpedm20/MemN2N-tensorflow>

* pip install future
* pip install progress
* To train a model with 6 hops and memory size of 100, run the following command:
* python main.py --nhop 6 --mem\_size 100
* python main.py –help
* (Optional) If you want to see a progress bar, install progress with pip:
* python main.py --nhop 6 --mem\_size 100 --show True
* After training is finished, you can test and validate with:
* python main.py --is\_test True --show True

1. code: <https://appliedmachinelearning.blog/2019/05/02/building-end-to-end-memory-network-for-question-answering-system-on-babi-facebook-data-set-python-keras-part-2/>