# 13个Python自然语言处理的深度学习框架

作者：Olga Davydova

本文讨论了Python编程语言中最流行的用于自然语言处理（NLP）的神经网络框架和库。我们也看看这些工具的现有例子。

A是特别创造的。每个带有加号的单元格都包含指向NLP任务和网络类型透视图中框架使用示例的链接。



### 链子

由2014成立的日本公司开发的，是一种强大的、灵活的、直观的基于Python的神经网络框架，采用了一种方案。它存储计算历史，而不是编程逻辑。Chainer支持计算和。根据和发布的框架已经应用于情感分析、机器翻译、语音识别、问答等，使用不同类型的神经网络，如卷积网络、递归网络和序列到序列模型。



### 深度学习4J

是一个深入学习Java编程库，但它也有一个PythonAPI，Keras将在下面介绍。它的主要特点是分布式cpu和gpu、迭代归约并行训练、微服务架构自适应。使该工具能够解决文本挖掘问题。中讨论了词性标记、依赖性分析和用于创建单词嵌入的word2vec。

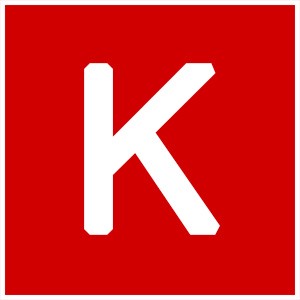
### 深度NL

是另一个专门为自然语言处理而创建的神经网络Python库。它为词性标注、命名实体识别、语义角色标注（使用卷积神经网络）和单词嵌入创建提供了工具。



### 双网

是由许多其他人开发的工具。它支持C++和Python语言，可以在CPU上运行，也可以在GPU上运行。Dynet是基于网络结构的动态声明。这个工具被用来为NLP问题创建优秀的系统，包括句法分析、机器翻译、形态屈折和许多其他问题。



### 路缘石

是一个基于高级神经网络的Python API，运行在CPU或GPU上。它支持卷积和递归网络，可以运行在或的顶部。主要关注点是让用户能够快速进行实验。在分类、文本生成和摘要、标记、解析、机器翻译、语音识别等方面有许多使用Keras的例子。

### Nlpnet公司

Erick Rocha Fonseca也是一个基于神经网络的用于NLP任务的Python库。卷积网络使用户能够执行词性标注、语义角色标注和依赖性分析。大多数建筑都是独立于语言的。



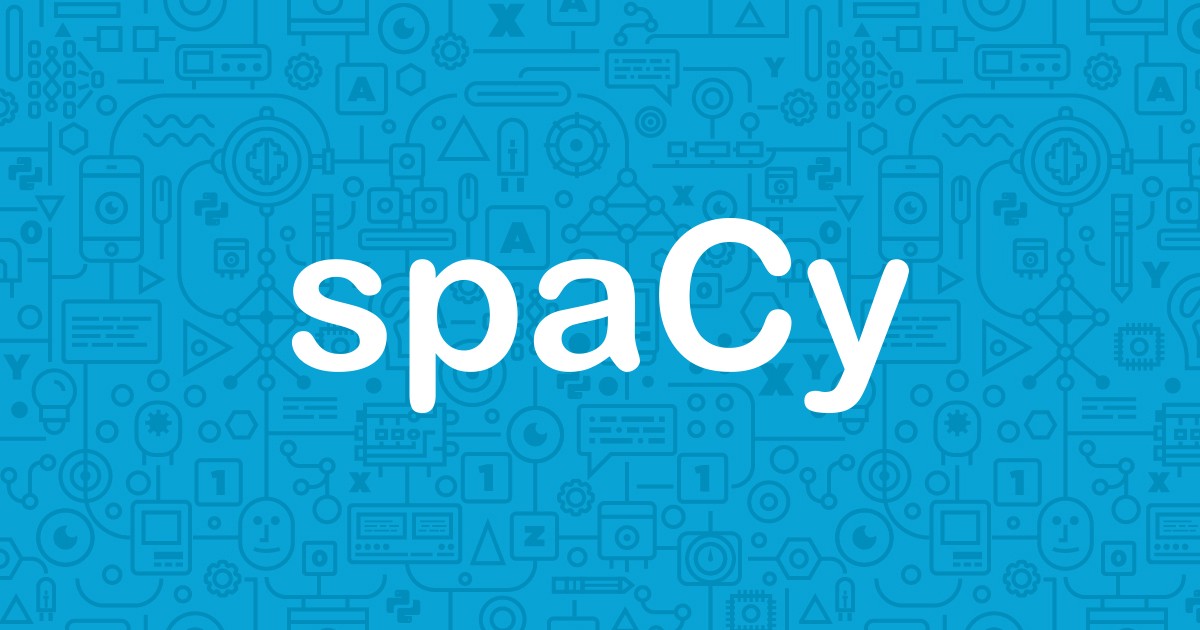
### 开放式

是一个Python机器翻译工具，在MIT许可下工作，依赖于库。该系统显示了效率和最先进的翻译准确性，并被许多翻译供应商使用。它还集成了文本摘要、语音识别和图像到文本的转换块。



### 火把

是一个快速和灵活的神经网络框架，具有必要的范例。它在基于磁带的自动加载系统上建立神经网络，并提供具有强GPU加速度的张量计算。递归神经网络主要用于PyTorch中的机器翻译、分类、文本生成、标记和其他NLP任务。



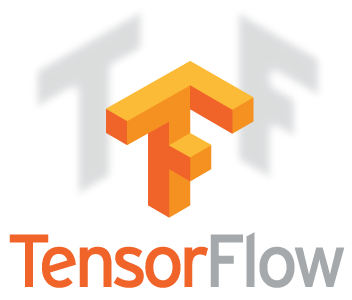
### 痉挛

开发人员称之为世界上速度最快的系统。他们还确认，他们的工具是为深度学习准备课文的最佳方式。Spacy与著名的Python库（如gensim、Keras、TensorFlow和scikit learn）配合使用非常好。《图书馆》的作者斯帕西说，斯帕西的使命是使最前沿的NLP实用化和普遍可用。比较表中给出了文本分类、命名实体识别、词性标注、依存度分析等实例。



### 斯坦福大学的CoreNLP

是一种灵活、快速、现代的语法分析工具，它为包括Python在内的大多数通用编程语言提供API。它还可以作为一个简单的web服务运行。如上所述，该框架具有部分语音（POS）标记器、命名实体识别器（NER）、解析器、共指解析系统、情感分析、自举模式学习和开放信息提取工具。



### 张量流

该团队在2015年开发并发布了它，用于研究目的。现在许多公司，如空客、英特尔、IBM、Twitter和其他公司在生产规模上使用TensorFlow。该体系结构是灵活的，因此可以在CPU或GPU上执行计算。主要概念是流图的使用。图的节点反映数学运算，而边表示它们之间通信的多维数据数组（张量）。TensorFlow最著名的NLP应用之一是。其他应用包括文本分类和摘要、语音识别、标记等。

### t学习

由于Tensorflow是一个低级API，因此创建了许多高级API以在其上运行，从而使用户体验更快、更容易理解。是在CPU和GPU上运行的工具之一。它有一个特殊的图形可视化工具，包含有关权重、渐变、激活等的详细信息。该库已经用于情感分析、文本生成和命名实体识别。它允许用户使用卷积神经网络和递归神经网络（LSTM）。



### 西雅娜

是一个数值计算Python库，它允许用户创建自己的机器学习模型。像Keras这样的许多框架都是建立在ano之上的。有用于机器翻译、语音识别、单词嵌入和文本分类的工具。查看中Theano的应用程序。

### Summary

In this paper, we described neural network supporting Python tools for natural language processing. These tools are Chainer, Deeplearning4j, Deepnl, Dynet, Keras, Nlpnet, OpenNMT, PyTorch, SpaCy, Stanford’s CoreNLP, TensorFlow, TFLearn, and Theano. A table lets readers easily compare the frameworks discussed above.

### Resources

1. <https://chainer.org/>
2. <http://learningsys.org/papers/LearningSys_2015_paper_33.pdf>
3. <https://deeplearning4j.org/>
4. <http://www.aclweb.org/anthology/W15-1515>
5. <https://github.com/attardi/deepnl>
6. <https://github.com/clab/dynet>
7. <https://arxiv.org/pdf/1701.03980.pdf>
8. <https://keras.io/>
9. <https://github.com/erickrf/nlpnet>
10. <http://nilc.icmc.usp.br/nlpnet/>
11. <http://opennmt.net/>
12. <http://opennmt.net/OpenNMT/applications/>
13. <http://pytorch.org/about/>
14. <https://spacy.io/>
15. <https://stanfordnlp.github.io/CoreNLP/>
16. <https://www.tensorflow.org/>
17. <http://tflearn.org/>
18. <https://github.com/Theano/Theano>
19. <https://github.com/odashi/chainer_nmt>

### Additional resources

<https://arxiv.org/pdf/1703.04783.pdf>

<https://github.com/chainer/chainer/tree/master/examples/word2vec>

<https://github.com/chainer/chainer/tree/master/examples/sentiment>

<https://github.com/marevol/cnn-text-classification>

<https://github.com/butsugiri/chainer-rnn-ner>

<https://github.com/khanhptnk/seq2seq-chainer>

<https://github.com/kenkov/seq2seq>

<https://github.com/chainer/chainer/tree/master/examples/ptb>

<https://github.com/masashi-y/chainer-parser>

<https://cs.stanford.edu/~danqi/papers/emnlp2014.pdf>

<http://learningsys.org/papers/LearningSys_2015_paper_33.pdf>

<https://arxiv.org/pdf/1611.01604.pdf>

<https://deeplearning4j.org/nlp>

<https://github.com/deeplearning4j/dl4j-examples/blob/master/dl4j-examples/src/main/java/org/deeplearning4j/examples/nlp/paragraphvectors/ParagraphVectorsClassifierExample.java>

<https://deeplearning4j.org/word2vec>

<https://github.com/deeplearning4j/dl4j-examples/blob/master/dl4j-examples/src/main/java/org/deeplearning4j/examples/recurrent/character/GravesLSTMCharModellingExample.java>

<http://www.aclweb.org/anthology/W15-1515>

<https://github.com/attardi/deepnl>

<https://github.com/attardi/deepnl/blob/master/deepnl/pos_tagger.py>

<https://github.com/attardi/deepnl/blob/master/deepnl/networkconv.pyx>

<https://github.com/attardi/deepnl/blob/master/deepnl/ner_tagger.py>

<https://github.com/attardi/deepnl/blob/master/deepnl/classifier.pyx>

<https://github.com/attardi/deepnl/blob/master/deepnl/sentiwords.pyx>

<http://www.aclweb.org/anthology/W15-1515>

<https://github.com/attardi/deepnl/blob/master/deepnl/embeddings.py>

<https://github.com/attardi/deepnl/blob/master/deepnl/tagger.pyx>

<https://github.com/attardi/deepnl/blob/master/deepnl/networkseq.pyx>

<https://github.com/attardi/deepnl/blob/master/deepnl/extractors.pyx>

<https://github.com/clab/dynet>

<https://arxiv.org/pdf/1701.03980.pdf>

<https://github.com/neubig/lamtram>

<https://github.com/bplank/bilstm-aux>

<http://phontron.com/slides/emnlp2016-dynet-tutorial-part1.pdf>

<https://github.com/toru34/kim_emnlp_2014>

<https://github.com/roeeaharoni/dynmt-py>

<http://phontron.com/slides/emnlp2016-dynet-tutorial-part2.pdf>

<http://dynet.readthedocs.io/en/latest/tutorials_notebooks/RNNs.html>

<https://github.com/clab/lstm-parser>

<https://github.com/clab/joint-lstm-parser>

<https://github.com/neubig/modlm>

<https://github.com/odashi/nmtkit>

<https://github.com/lvapeab/nmt-keras>

<https://chsasank.github.io/spoken-language-understanding.html>

<https://github.com/igormq/asr-study>

<https://github.com/llSourcell/How_to_make_a_text_summarizer>

<https://github.com/fchollet/keras/blob/master/examples/imdb_lstm.py>

<https://github.com/fchollet/keras/blob/master/examples/lstm_text_generation.py>

<https://github.com/farizrahman4u/seq2seq>

<https://github.com/fchollet/keras/blob/master/examples/imdb_cnn.py>

<https://github.com/udibr/headlines>

<https://github.com/wolet/s2s-dependency-parsers>

<https://github.com/0xnurl/keras_character_based_ner>

<https://github.com/fchollet/keras/blob/master/examples/imdb_bidirectional_lstm.py>

<https://github.com/fchollet/keras/blob/master/examples/reuters_mlp.py>

<https://github.com/codekansas/keras-language-modeling>

[https://link.springer.com/content/pdf/10.1186%2Fs13173-014-0020-x.pdf](https://link.springer.com/content/pdf/10.1186/s13173-014-0020-x.pdf)

<http://nilc.icmc.sc.usp.br/nlpnet/models.html#word-embeddings-portuguese>

<http://nilc.icmc.sc.usp.br/nlpnet/models.html#pos-portuguese>

<http://nilc.icmc.sc.usp.br/nlpnet/models.html#srl-portuguese>

<http://nilc.icmc.sc.usp.br/nlpnet/models.html#dependency-and-pos-english>

<https://github.com/erickrf/nlpnet/blob/master/nlpnet/taggers.py>

<https://github.com/erickrf/nlpnet/blob/master/nlpnet/networkconv.pyx>

<https://github.com/erickrf/nlpnet/blob/master/nlpnet/networkdependencyconv.pyx>

<http://www.aclweb.org/anthology/W15-1508>

<https://github.com/erickrf/nlpnet>

<http://nilc.icmc.usp.br/nlpnet/>

<http://opennmt.net/OpenNMT/applications/#machine-translation>

<http://opennmt.net/OpenNMT/applications/#summarization>

<http://opennmt.net/OpenNMT/applications/#speech-recognition>

<http://opennmt.net/OpenNMT/applications/#sequence-tagging>

<http://opennmt.net/OpenNMT/applications/#language-modelling>

<http://opennmt.net/OpenNMT/training/embeddings/>

<https://arxiv.org/pdf/1701.02810.pdf>

<http://opennmt.net/OpenNMT/applications/>

<http://pytorch.org/about/>

<http://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html>

[http://pytorch.org/tutorials/beginner/nlp/sequence\_models\_tutorial.html#](http://pytorch.org/tutorials/beginner/nlp/sequence_models_tutorial.html)

<http://pytorch.org/tutorials/beginner/nlp/word_embeddings_tutorial.html>

<http://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial.html>

<https://github.com/spro/practical-pytorch/blob/master/char-rnn-generation/char-rnn-generation.ipynb>

<https://github.com/spro/practical-pytorch/blob/master/conditional-char-rnn/conditional-char-rnn.ipynb>

<https://spacy.io/>

<https://spacy.io/docs/usage/pos-tagging>

<https://spacy.io/docs/usage/word-vectors-similarities>

<https://spacy.io/docs/usage/entity-recognition>

<https://spacy.io/docs/usage/dependency-parse>

<https://spacy.io/docs/usage/deep-learning>

<https://explosion.ai/blog/spacy-deep-learning-keras>

<https://stanfordnlp.github.io/CoreNLP/>

<http://apps.cs.utexas.edu/tech_reports/reports/tr/TR-2222.pdf>

<https://nlp.stanford.edu/projects/mt.shtml>

<https://github.com/Lynten/stanford-corenlp>

<https://github.com/stanfordnlp/treelstm>

<https://arxiv.org/pdf/1609.08409.pdf>

<https://nlp.stanford.edu/sentiment/>

<https://www.tensorflow.org/>

<https://github.com/tensorflow/nmt>

<https://arxiv.org/pdf/1609.08144.pdf>

<https://github.com/mrahtz/tensorflow-pos-tagger>

<https://github.com/pannous/tensorflow-speech-recognition>

<https://www.tensorflow.org/tutorials/word2vec>

<https://github.com/monikkinom/ner-lstm>

<https://github.com/dennybritz/cnn-text-classification-tf>

<https://research.googleblog.com/2016/08/text-summarization-with-tensorflow.html>

<https://github.com/tensorflow/models/tree/master/research/textsum>

<https://www.tensorflow.org/tutorials/recurrent>

<http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/>

<http://jrmeyer.github.io/tutorial/2016/02/01/TensorFlow-Tutorial.html>

<http://tflearn.org/>

<https://github.com/dhwajraj/NER-RNN>

<https://github.com/tflearn/tflearn/blob/master/examples/nlp/cnn_sentence_classification.py>

<https://github.com/tflearn/tflearn/blob/master/examples/nlp/bidirectional_lstm.py>

<https://github.com/tflearn/tflearn/blob/master/examples/nlp/lstm.py>

<https://github.com/tflearn/tflearn/blob/master/examples/nlp/seq2seq_example.py>

<https://github.com/tflearn/tflearn/blob/master/examples/nlp/lstm_generator_shakespeare.py>

<https://github.com/tflearn/tflearn/blob/master/examples/nlp/lstm_generator_cityname.py>

<https://github.com/EdinburghNLP/nematus>

<https://github.com/ZhangAustin/Deep-Speech>

<https://github.com/llSourcell/How_to_make_a_text_summarizer>

<http://deeplearning.net/tutorial/rnnslu.html>

<https://raberrytv.wordpress.com/2016/12/26/efficient-embeddings-with-theano/>

<https://github.com/yoonkim/CNN_sentence>

<https://deeplearning4j.org/convolutionalnets.html>

<https://deeplearning4j.org/usingrnns>

<https://github.com/neulab/xnmt>

<https://github.com/memeda/sequence-labeling-by-nn>

<https://cs.umd.edu/~miyyer/pubs/2017_acl_dynsp.pdf>

<http://ben.bolte.cc/blog/2016/language.html>

<http://pyvideo.org/pydata-carolinas-2016/deep-language-modeling-for-question-answering-using-keras.html>

<https://github.com/chartbeat-labs/textacy>

<http://iamaaditya.github.io/2016/04/visual_question_answering_demo_notebook>

<https://github.com/hans/corenlp-summarizer>

<https://nlp.stanford.edu/software/relationExtractor.html>

<https://github.com/spiglerg/RNN_Text_Generation_Tensorflow>

<https://github.com/paarthneekhara/neural-vqa-tensorflow>

<https://github.com/DeepRNN/visual_question_answering>

<https://github.com/llSourcell/How_to_do_Sentiment_Analysis>

<https://github.com/Lasagne/Recipes/blob/master/examples/lstm_text_generation.py>

<https://github.com/glample/tagger>

<https://github.com/Sentimentron/Dracula>

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.646.4491&rep=rep1&type=pdf>

<https://github.com/hiroki13/neural-semantic-role-labeler>

<https://github.com/carpedm20/hali>

<https://github.com/saltypaul/Seq2Seq-Chatbot>