Home

8 Excellent Pretrained Models to get you Started with Natural Language Processing (NLP)

Pranav Dar — March 18, 2019

Deep Learning Intermediate Listicle NLP Python Resource

Introduction

Natural Language Processing (NLP) applications have become ubiquitous these days. I seem to stumble across websites and applications regularly that are leveraging NLP in one form or another. In short, this is a wonderful time to be involved in the NLP domain.

This rapid increase in NLP adoption has happened largely thanks to the concept of transfer learning enabled through pretrained models. Transfer learning, in the context of NLP, is essentially the ability to train a model on one dataset and then adapt that model to perform different NLP functions on a different dataset.

This breakthrough has made things incredibly easy and simple for everyone, especially folks who don't have the time or resources to build NLP models from scratch. It's perfect for beginners as well who want to learn or transition into NLP.

Why use pretrained models?

- The author(s) has already put in the effort to design a benchmark model for you! Instead of building a model from scratch to solve a similar NLP problem, we can use that pretrained model on our own NLP dataset
- A bit of fine-tuning will be required but it saves us a ton of time and computational resources

In this article, I have showcased the top pretrained models you can use to start your NLP journey and replicate the state-of-the-art research in this field. You can check out my article on the top pretrained models in Computer Vision here.

Pretrained NLP Models Covered in this Article

I have classified the pretrained models into three different categories based on their application:

- **Multi-Purpose NLP Models**
  - ULMFiT
  - Transformer
  - Google's BERT
  - Transformer-XL
  - OpenAI's GPT-2
- **Word Embeddings**
  - ELMo
  - Flair
- **Other Pretrained Models**
  - StanfordNLP

Multi-Purpose NLP Models

Multi-purpose models are the talk of the NLP world. These models power the NLP applications we are excited about – machine translation, question answering systems, chatbots, sentiment analysis, etc. A core component of these multi-purpose NLP models is the concept of language modelling.

In simple terms, the aim of a language model is to predict the next word or character in a sequence. We'll understand this as we look at each model here.

If you're a NLP enthusiast, you're going to love this section. Now, let's dive into 5 state-of-the-art multi-purpose NLP model frameworks. I have provided links to the research paper and pretrained models for each model. Go ahead and explore them!

ULMFiT

ULMFiT was proposed and designed by fast.ai's Jeremy Howard and DeepMind's Sebastian Ruder. You could say that ULMFiT was the release that got the transfer learning party started last year.

As we have covered in this article, ULMFiT achieves state-of-the-art results using novel NLP techniques. This method involves fine-tuning a pretrained language model, trained on the Wikitext 103 dataset, to a new dataset in such a manner that it does not forget what it previously learned.

ULMFiT outperforms numerous state-of-the-art on text classification tasks. What I liked about ULMFiT is that it needs very few examples to produce these impressive results. Makes it easier for folks like you and me to understand and implement it on our machines!

In case you were wondering, ULMFiT stands for Universal Language Model Fine-Tuning. The word 'Universal' is quite apt here – the framework can be applied to almost any NLP task.

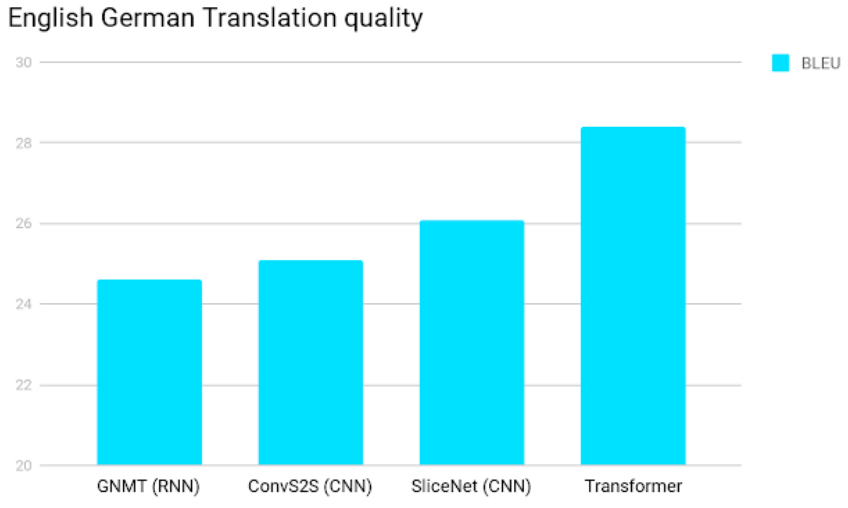
Resources to learn and read more about ULMFiT:

- Tutorial on Text Classification (NLP) using ULMFiT and fastai Library in Python
- Pretrained models for ULMFiT
- Research Paper

Transformer

The Transformer architecture is at the core of almost all the recent major developments in NLP. It was introduced in 2017 by Google. Back then, recurrent neural networks (RNN) were being used for language tasks, like machine translation and question answering systems.

This Transformer architecture outperformed both RNNs and CNNs (convolutional neural networks). The computational resources required to train models were reduced as well. A win-win for everyone in NLP. Check out the below comparison:



As per Google, Transformer "applies a self-attention mechanism which directly models relationships between all words in a sentence, regardless of their respective position". It does so using a fixed-sized context (aka the previous words). Too complex to get? Let's take an example to simplify this.

"She found the shells on the bank of the river." The model needs to understand that "bank" here refers to the shore and not a financial institution. Transformer understands this in a single step. I encourage you to read the full paper I have linked below to gain an understanding of how this works. It will blow your mind.

The below animation wonderfully illustrates how Transformer works on a machine translation task:

Google released an improved version of Transformer last year called Universal Transformer. There's an even newer and more intuitive version, called Transformer-XL, which we will cover below.

Resources to learn and read more about Transformer:

- Google's official blog post
- Pretrained models for Transformer
- Research Paper

Google's BERT

The BERT framework has been making waves ever since Google published their results, and then open sourced the code behind it. We can debate whether this marks "a new era in NLP", but there's not a shred of doubt that BERT is a very useful framework that generalizes well to a variety of NLP tasks.

BERT, short for **B**idirectional **E**ncoder **R**epresentations, considers the context from both sides (left and right) of a word. All previous efforts considered one side of a word at a time – either the left or the right. This bidirectionality helps the model gain a much better understanding of the context in which the word(s) was used. Additionally, BERT is designed to do multi-task learning, that is, it can perform different NLP tasks simultaneously.

BERT is the first unsupervised, deeply bidirectional system for pretraining NLP models. It was trained using only a plain text corpus.

At the time of its release, BERT was producing state-of-the-art results on 11 Natural Language Processing (NLP) tasks. Quite a monumental feat! You can train your own NLP model (such as a question-answering system) using BERT in just a few hours (on a single GPU).

Resources to learn and read more about BERT:

- Google's official blog post
- Pretrained models for BERT
- Research Paper

Google's Transformer-XL

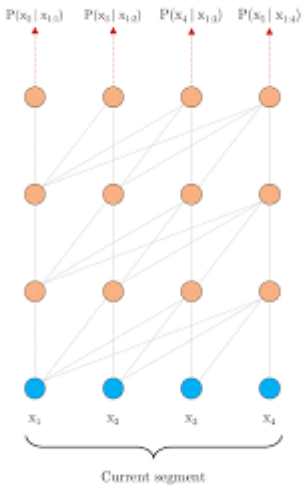
This release by Google could potentially be a very important one in the long-term for NLP. This concept could become a bit tricky if you're a beginner so I encourage you to read it a few times to grasp it. I have also provided multiple resources below this

Picture this – you're halfway through a book and suddenly a word or sentence comes up that was referred to at the start of the book. Now, you or I can recall what it was. But a machine, understandably, struggles to model long-term dependency.

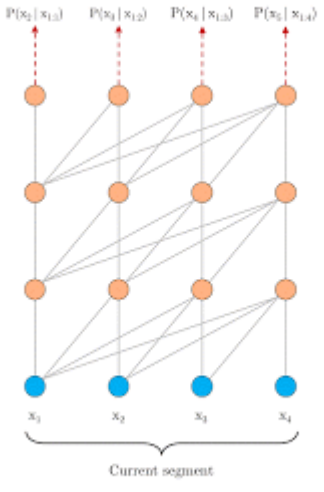
One way to do this, as we saw above, is by using Transformers. But they are implemented with a fixed-length context. In other words, there's not much flexibility to go around if you use this approach.

Transformer-XL bridges that gap really well. Developed by the Google AI team, it is a novel NLP architecture that helps machines understand context beyond that fixed-length limitation. **Transformer-XL is up to 1800 times faster than a typical Transformer.**

You'll understand this difference through the below 2 GIFs released by Google:



Vanilla Transformer



Transformer-XL

Transformer-XL, as you might have predicted by now, achieves new state-of-the-art results on various language modeling benchmarks/datasets. Here's a small table taken from their page illustrating this:

Method	enwiki8	text8	One Billion Word	WT-103	PTB (w/o finetuning)
Previous Best	1.06	1.13	23.7	20.5	55.5
Transformer-XL	**0.99**	**1.08**	**21.8**	**18.3**	**54.5**

The Transformer-XL GitHub repository, linked above and mentioned below, contains the code in both PyTorch and TensorFlow.

Resources to learn and read more about Transformer-XL :

- Research Paper

OpenAI's GPT-2

Now, this is a pretty controversial entry. A few people might argue that the release of GPT-2 was a marketing stunt by OpenAI. I certainly understand where they're coming from. However, I believe it's important to still at least try out the code OpenAI has released.

First, some context for those who are not aware what I'm talking about. OpenAI penned a blog post (link below) in February where they claimed to have designed a NLP model, called GPT-2, that was so good that they couldn't afford to release the full version for fear of malicious use. That certainly got the community's attention.

GPT-2 was trained to predict the next occurring word in 40GB of internet text data. This framework is also a transformer-based model trained on a dataset of 8 million web pages. The results they have published on their site are nothing short of astounding. The model is able to weave an entirely legible story based on a few sentences we input. Check out this example:

SYSTEM PROMPT (HUMAN-WRITTEN)	*In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*
MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)	The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow. Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Incredible, right?

The developers have released a much smaller version of GPT-2 for researchers and engineers to test. The original model has 1.5 billion parameters – the open source sample model has 117 million.

Resources to learn and read more about GPT-2:

- Research paper

Word Embeddings

Most of the machine learning and deep learning algorithms we use are incapable of working directly with strings and plain text. These techniques require us to convert text data into numbers before they can perform any task (such as regression or classification).

So in simple terms, word embeddings are the text blocks that are converted into numbers for performing NLP tasks. A word bmbedding format generally tries to map a word using a dictionary to a vector.

You can get a much more in-depth explanation of word embeddings, its different types, and how to use them on a dataset in the below article. If you are not familiar with the concept, I consider this guide a must-read:

- An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec

In this section, we'll look at two state-of-the-art word embeddings for NLP. I have also provided tutorial links so you can get a practical understanding of each topic.

ELMo

No, this ELMo isn't the (admittedly awesome) character from Sesame Street. But this ELMo, short for Embeddings from Language Models, is pretty useful in the context of building NLP models.

ELMo is a novel way of representing words in vectors and embeddings. These ELMo word embeddings help us achieve state-of-the-art results on multiple NLP tasks, as shown below:

Let's take a moment to understand how ELMo works. Recall what we discussed about bidirectional language models earlier. Taking a cue from this article, "ELMo word vectors are computed on top of a two-layer bidirectional language model (biLM). This biLM model has two layers stacked together. Each layer has 2 passes — forward pass and backward pass:

ELMo word representations consider the full input sentence for calculating the word embeddings. So, the term “read” would have different ELMo vectors under different context. A far cry from the older word embeddings when the same vector would be assigned to the word “read” regardless of the context in which it was used.

Resources to learn and read more about ELMo:

- Step-by-Step NLP Guide to Learn ELMo for Extracting Features from Text
- GitHub repository for pretrained models
- Research Paper

Flair

Flair is not exactly a word embedding, but a combination of word embeddings. We can call Flair more of a NLP library that combines embeddings such as GloVe, BERT, ELMo, etc. The good folks at Zalando Research developed and open-sourced Flair.

The team has released several pretrained models for the below NLP tasks:

- **Name-Entity Recognition (NER)**
- **Parts-of-Speech Tagging (PoS)**
- **Text Classification**
- **Training Custom Models**

Not convinced yet? Well, this comparison table will get you there:

Task	Language	Dataset	Flair	Previous best
Named Entity Recognition	English	Conll-03	**93.09** (F1)	*92.22 (Peters et al., 2018)*
Named Entity Recognition	English	Ontonotes	**89.71** (F1)	*86.28 (Chiu et al., 2016)*
Emerging Entity Detection	English	WNUT-17	**50.20** (F1)	*45.55 (Aguilar et al., 2018)*
Part-of-Speech tagging	English	WSJ	**97.85**	*97.64 (Choi, 2016)*
Chunking	English	Conll-2000	**96.72** (F1)	*96.36 (Peters et al., 2017)*
Named Entity Recognition	German	Conll-03	**88.32** (F1)	*78.76 (Lample et al., 2016)*
Named Entity Recognition	German	Germeval	**84.65** (F1)	*79.08 (Hänig et al, 2014)*
Named Entity Recognition	Polish	PolEval-2018	**86.6** (F1) (Borchmann et al., 2018)	*85.1 (PolDeepNer)*

'Flair Embedding' is the signature embedding that comes packaged within the Flair library. It is powered by contextual string embeddings. You should go through this article to understand the core components that power Flair.

What I especially like about Flair is that it supports multiple languages. So many NLP releases are stuck doing English tasks. We need to expand beyond this if NLP is to gain traction globally!

Resources to learn and read more about Flair:

- Introduction to Flair for NLP: A Simple yet Powerful State-of-the-Art NLP Library
- Pretrained models for Flair

Other Pretrained Models

StanfordNLP

Speaking of expanding NLP beyond the English language, here's a library that is already setting benchmarks. The authors claim that StanfordNLP supports over 53 languages – that certainly got our attention!

Our team was among the first to work with the library and publish the results on a real-world dataset. We played around with it and found that StanfordNLP truly does open up a lot of possibilities of applying NLP techniques on non-English languages. like Hindi, Chinese and Japanese.

StanfordNLP is a collection of pretrained state-of-the-art NLP models. These models aren't just lab tested – they were used by the authors in the CoNLL 2017 and 2018 competitions. All the pretrained NLP models packaged in StanfordNLP are built on PyTorch and can be trained and evaluated on your own annotated data.

    - Tokenization
    - Multi-word token (MWT) expansion
    - Lemmatization
    - Parts-of-speech (POS) and morphological feature tagging
    - Dependency Parsing
- A stable officially maintained Python interface to CoreNLP

Resources to learn and read more about StanfordNLP:

- Introduction to StanfordNLP: An Incredible State-of-the-Art NLP Library for 53 Languages (with Python code)
- Pretrained models for StanfordNLP

End Notes

This is by no means an exhaustive list of pretrained NLP models. There are a lot more available and you can check out a few of them on this site.

Here are a couple of useful resources for learning NLP:

- **Natural Language Processing using Python course**
- **Certified Program: NLP for Beginners**
- **Collection of articles on Natural Language Processing (NLP)**

I would love to hear your thoughts on this list. Have you used any of these pretrained models before? Or you have perhaps explored other options? Let me know in the comments section below – I will be happy to check them out and add them to this list.

deep learning Natural language processing NLP pretrained models python

About the Author

Pranav Dar

Senior Editor at Analytics Vidhya. Data visualization practitioner who loves reading and delving deeper into the data science and machine learning arts. Always looking for new ways to improve processes using ML and AI.

> 5 Amazing Deep Learning Frameworks Every Data Scientist Must Know! (with Illustrated Infographic)

> DataHack Radio #20: Building Interpretable Machine Learning Models with Christoph Molnar

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name*

Email*

Website

Notify me of follow-up comments by email.

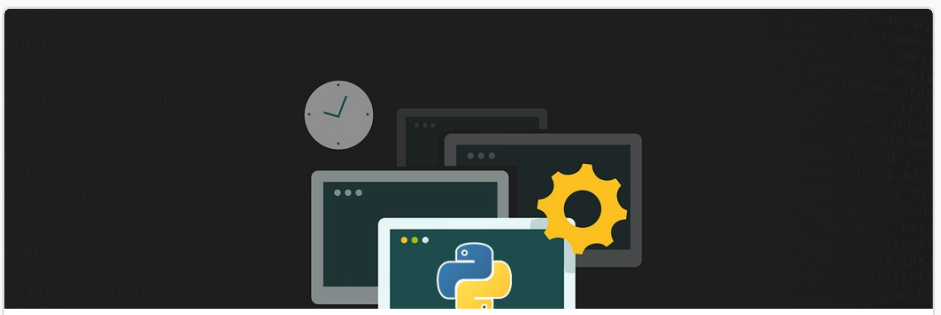
Notify me of new posts by email.

Submit

Top Resources

SQL: A Full Fledged Guide from Basics to Advance Level

Harsh Kulkarni - JAN 19, 2022



Python Tutorial: Working with CSV file for Data Science

Harika Bonthu - AUG 21, 2021

3 Interesting Python Projects With Code for Beginners!

Commonly used Machine Learning Algorithms (with Python

Gaurav Sharma - JUL 18, 2021

Sunil Ray - SEP 09, 2017