Yes, there are many other significant and widely used Python libraries and frameworks in the field of Natural Language Processing and Deep Learning for NLP. Here are a few prominent ones:

1. **Hugging Face Transformers**
   * **Introduced:** The Hugging Face transformers library gained significant traction starting in **2019**, following the widespread adoption of Transformer models (like BERT, GPT-2). It quickly became the go-to library for accessing and utilizing pre-trained state-of-the-art NLP models.
   * **Focus:** Provides thousands of pre-trained models for tasks like text classification, named entity recognition, question answering, summarization, generation, and more. It abstracts away the complexity of working with deep learning frameworks (like PyTorch and TensorFlow) directly for these models.
2. **AllenNLP**
   * **Introduced:** AllenNLP was released by the Allen Institute for AI (AI2) in **2018**.
   * **Focus:** Built on PyTorch, AllenNLP is a research-oriented NLP library that provides high-level abstractions for common NLP tasks, making it easier to build and experiment with new deep learning models for text. It's known for its structured approach to defining models, data readers, and training loops.
3. **TextBlob**
   * **Introduced:** TextBlob was initially released around **2013-2014**.
   * **Focus:** It's a Python library for processing textual data, providing a simple API for common NLP tasks like part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. It's often favored for its ease of use for quick prototyping and simpler tasks, sitting on top of NLTK.
4. **Flair**
   * **Introduced:** Flair was developed by Zalando Research and first released around **2018**.
   * **Focus:** It's a powerful NLP framework built on PyTorch, particularly known for its state-of-the-art results in sequence labeling tasks (like Named Entity Recognition and Part-of-Speech tagging) using contextual string embeddings. It also offers text classification, word embeddings, and more.

These libraries, along with NLTK, SpaCy, and Gensim, form the backbone of Python's rich ecosystem for natural language processing, each serving slightly different needs from research and development to production deployment and ease of use.

Let's look at the approximate introduction or first major release years for these popular NLP libraries and techniques:

1. **NLTK (Natural Language Toolkit)**
   * **Introduced:** NLTK's development began in **2001** by Steven Bird and Edward Loper. The first stable release was around **2002-2003**. It's one of the oldest and most established NLP libraries in Python.
2. **SpaCy**
   * **Introduced:** SpaCy was first released in **2015** by explosion.ai (Matthew Honnibal and Ines Montani). It was designed for production use, focusing on speed and efficiency.
3. **Gensim word2vec**
   * **Introduced:** The Word2Vec algorithm itself was developed by Google's Tomas Mikolov and his team and published in **2013**. Gensim, an open-source library for topic modeling and natural language processing, quickly implemented Word2Vec. Gensim's implementation of Word2Vec became widely available and popular starting in **2013** itself, shortly after Google's papers were released.

The GPT (Generative Pre-trained Transformer) series of models has been exclusively developed by **OpenAI**.

Here's the sequence of their major public releases:

1. **GPT-1 (Generative Pre-trained Transformer 1)**
   * **Introduced:** 2018
   * **Developer:** OpenAI
   * **Significance:** Marked OpenAI's first foray into large-scale, pre-trained language models using the Transformer architecture. It demonstrated the power of unsupervised pre-training on a diverse text corpus.
2. **GPT-2 (Generative Pre-trained Transformer 2)**
   * **Introduced:** 2019
   * **Developer:** OpenAI
   * **Significance:** Known for its impressive text generation capabilities and controversial staggered release due to concerns about misuse. It significantly scaled up the model size and training data compared to GPT-1.
3. **GPT-3 (Generative Pre-trained Transformer 3)**
   * **Introduced:** 2020
   * **Developer:** OpenAI
   * **Significance:** A monumental leap in scale (up to 175 billion parameters). It demonstrated remarkable few-shot learning abilities, performing well on tasks it wasn't explicitly trained for, just by seeing a few examples in the prompt. This model truly popularized the concept of LLMs.
4. **GPT-3.5 Series**
   * **Introduced:** 2022 (e.g., text-davinci-003, the base for early ChatGPT versions)
   * **Developer:** OpenAI
   * **Significance:** These are refined and fine-tuned versions of GPT-3, often using techniques like Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF). This series led to models that were exceptionally good at following instructions and engaging in conversational dialogue, most famously powering **ChatGPT**.
5. **GPT-4**
   * **Introduced:** 2023
   * **Developer:** OpenAI
   * **Significance:** A major advancement over GPT-3.5, showcasing improved reasoning, creativity, and the ability to handle much longer and more complex prompts. It's also notable for its **multimodal capabilities**, being able to accept image inputs in addition to text (though this feature is not yet broadly available for all users).

OpenAI continues to research and develop next-generation GPT models.