**AST (абстрактное синтаксическое дерево) -** представляет собой внутреннее представление исходного кода, которое можно анализировать и модифицировать программно (больше используется как шаблон для изменения кода).

**Codegen** – обратное преобразование из AST в адекватный вид с применением различных преобразований кода.

**spaСy** - библиотека для анализа описания (кода), есть функция выделения сущностей (тематик). Реализует NLP. Есть возможность дообучать уже существующие модели. Использования семантической близости для рекомендательных систем:

<https://spacy.io/> - официальный сайт

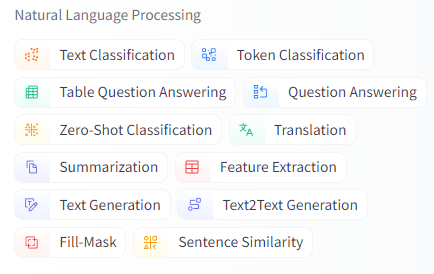
<https://course.spacy.io/en/> - ознакомительный курс

<https://habr.com/ru/articles/531940/> - статья на хабре

<https://prodi.gy/> - для удобного обучения выделения сущностей

# Transformers и Hugging Face - фреймворки решают необходимый нам спектр задач NLP: classification, NER (распознавание именованных сущностей), question answering, language modeling, summarization, translation, multiple choice, text generation.

<https://huggingface.co/models> - для формализации задачи (большое количество обученных моделей)



Tasks: https://huggingface.co/tasks

Text generation: https://huggingface.co/tasks/text-generation

Ps: некоторые **text generation** models можно использовать и для генерации кода. Например, **StarCoder**: <https://huggingface.co/spaces/bigcode/bigcode-playground>

Для более простого внедрения модели используется **pipeline:** https://huggingface.co/docs/transformers/main/en/main\_classes/pipelines

**Статья на хабре**: <https://habr.com/ru/articles/704592/>

**CodeBERT** - самое нужное! преобразование кода в текст, анализ кода на наличие ошибок, генерация кода по тексту.

<https://github.com/microsoft/CodeBERT>

<https://habr.com/ru/companies/sberbank/articles/737730/> - статься на хабре про генерацию комментариев

https://clay-atlas.com/us/blog/2021/10/27/codebert-introduction-2/

**NLTK** предлагает удобные инструменты для множества задач NLP: токенизация, стемминг, лемматизация, морфологический и синтаксический анализ, а также анализ настроений.

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

**Gensim** – анализ семантики.

<https://www.geeksforgeeks.org/nlp-gensim-tutorial-complete-guide-for-beginners/>

<https://pythonru.com/biblioteki/gensim>

**TextBlob** – еще одна библиотека nlp

<https://textblob.readthedocs.io/en/dev/>

**Black -** это Python-пакет, который автоматически форматирует код, приводя его внешний вид к стандарту [PEP 8](https://www.python.org/dev/peps/pep-0008/).

<https://pypi.org/project/black/>

https://peps.python.org/pep-0008/

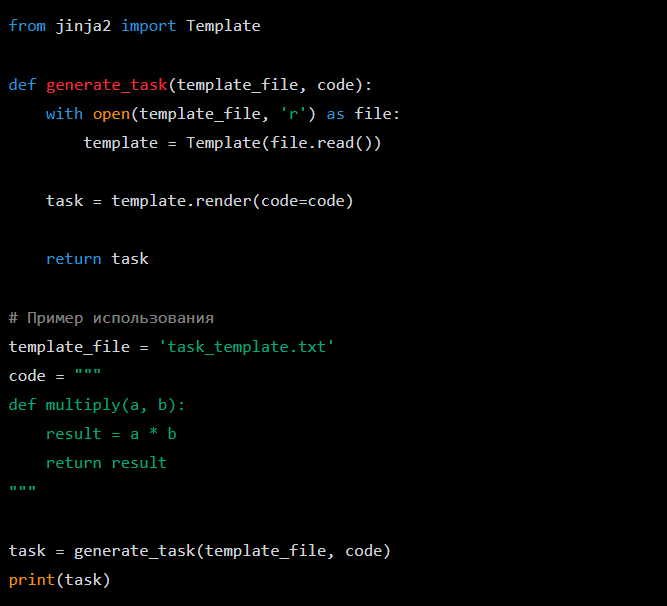
**PyTest –** библиотека для тестирования кода

<https://semaphoreci.com/community/tutorials/testing-python-applications-with-pytest>

<https://habr.com/ru/articles/502366/> - статься о библиотеках для распознавания именованных сущностей.

**План:**

1. Распознавание именованных сущностей (spaСy)
2. К найденным сущностям можно применить шаблонизатор (Jinja2). Таким образом, для определенного типа кода будет генерироваться нужные задачи



1. Если кода нет, то необходимо его сгенерировать. В таком случае, можно использовать библиотеку Transformers с моделью CodeBERT или подобными.

После генерации любого кода можно использовать библиотеку black для рефакторинга.