**AutoTM** — open-source библиотека автоматического тематического моделирования, предназначенная для решения задач в области обработки естественного языка.

Области применения:

* предварительный анализ данных для определения основных особенностей корпуса;
* получение векторных представлений текстов для дальнейшего использования в пайплайнах машинного обучения;
* тематическая разметка документов.

Общая структура библиотеки AutoTM представлена на рисунке 1. В ее состав входит набор модулей, подробно описанный далее в соответствующих пунктах настоящего раздела. Библиотека подразумевает два основных режима работы:

1. режим построения тематической модели (режим fit), который подразумевает подбор оптимальных гиперпараметров тематической модели ARTM для предоставленного корпуса документов с помощью алгоритма оптимизации, предоставляемого библиотекой и выбранного пользователем;
2. режим обслуживания (режим predict), который подразумевает оценку тематических смесей для документов из предоставленного на вход корпуса документов.

Следует отметить, что результатом работы библиотеки в первом режиме является обученная тематическая модель, построенная с помощью подобранных гиперпараметров и включающая в себя набор построенных тематик по предоставленному корпусу. Для этого пользователь обращается к модулю консольного клиента или напрямую к библиотеке из своего кода программы. В обоих случаях происходит вызов модуля предобработки и вызов модуля алгоритмов оптимизации, которые в свою очередь обращаются к модулю ядра и, при необходимости, к модулю распределенного выполнения.

Результатом работы во втором режиме - режиме обслуживания - является тематическая смесь для каждого предоставленного документа во входном корпусе. Полученные смеси могут быть использованы в широком спектре задач, начиная от разведочного анализа и заканчивая классификацией текстов

Здесь следует отметить, что документы входного корпуса могут не принадлежать корпусу, на котором изначально проводилось построение тематической модели. Например, для обучения подавалась только часть огромного изначального корпуса для экономии времени или в условиях ограниченности доступных вычислительных ресурсов. Еще одним примером для возникновения необходимости такого режима может служить ситуация в котором вновь поступающие документы не принадлежат изначальному корпусу, но поступают из того же источника, что и изначальный корпус, или имеют такую же природу генерации. Также, как и в предыдущем режиме пользователь обращается либо к консольному клиенту, либо напрямую к модулям библиотеки из своего кода программы. В обоих случаях происходит вызов модуля предобработки и вызов модуля обслуживания, которые использует ранее обученную ARTM модель для получения тематических смесей.


Рисунок 1 - Общая структура библиотеки AutoTM и связи между модулями

**Пример работы с библиотекой для получения тематической модели**

Для работы библиотеки необходимо подготовить документы в формате csv, где каждый документ расположен на отдельной строке.

После этого, пользователю необходимо вызвать функцию process\_dataset из autotm.preprocessing.text\_preprocessing и, затем, prepare\_all\_artifacts из autotm.preprocessing.dictionaries\_preparation для подготовки данных в формате для проведения процедуры моделирования.

Аргументы функции process\_dataset:

* fname - путь до датасета в формате csv который будет обработан
* сol\_to\_process - колонка с документами
* save\_path – путь, куда будут сохранены результаты обработки
* lang - язык датасета, поддерживаются русский (ru) и английский (en)
* min\_tokens\_count - минимальное количество токенов в документе для продолжения обработки и проведения тематического моделирования
* n\_cores - количество ядер для параллелизации

Аргументом функции prepare\_all\_artifacts является путь до сохраненного датасета (save\_path).

Путь до итоговых файлов подготовленного датасета подается на вход run\_algorithm из autotm.algorithms\_for\_tuning.genetic\_algorithm. genetic\_algorithm который скрывает всю внутреннюю реализацию оптимизационного пайплайна и обладает существенным набором настраиваемых параметров. Однако, для удобной работы пользователя достаточно указывать наименование датасета и путь до сохраненных подговленных на предыдущем этапе файлов.

**Полный базовый пример обучения модели представлен ниже:**

import time

import pandas as pd

from autotm.algorithms\_for\_tuning.genetic\_algorithm.genetic\_algorithm import (

 run\_algorithm,

)

from autotm.preprocessing.dictionaries\_preparation import prepare\_all\_artifacts

from autotm.preprocessing.text\_preprocessing import process\_dataset

PATH\_TO\_DATASET = "../data/sample\_corpora/sample\_dataset\_lenta.csv"

SAVE\_PATH = (

 "../data/processed\_sample\_corpora"

)

dataset = pd.read\_csv(PATH\_TO\_DATASET)

col\_to\_process = "text"

dataset\_name = "lenta\_sample"

lang = "ru"

min\_tokens\_num = 3

num\_iterations = 2

topic\_count = 10

exp\_id = int(time.time())

print(exp\_id)

use\_nelder\_mead\_in\_mutation = False

use\_nelder\_mead\_in\_crossover = False

use\_nelder\_mead\_in\_selector = False

train\_option = "offline"

if \_\_name\_\_ == "\_\_main\_\_":

 print("Stage 1: Dataset preparation")

 process\_dataset(

 PATH\_TO\_DATASET,

 col\_to\_process,

 SAVE\_PATH,

 lang,

 min\_tokens\_count=min\_tokens\_num,

 )

 prepare\_all\_artifacts(SAVE\_PATH)

 print("Stage 2: Tuning the topic model")

 best\_result = run\_algorithm(

 data\_path=SAVE\_PATH,

 dataset=dataset\_name,

 exp\_id=exp\_id,

 topic\_count=topic\_count,

 log\_file="./log\_file\_test.txt",

 num\_iterations=num\_iterations,

 use\_nelder\_mead\_in\_mutation=use\_nelder\_mead\_in\_mutation,

 use\_nelder\_mead\_in\_crossover=use\_nelder\_mead\_in\_crossover,

 use\_nelder\_mead\_in\_selector=use\_nelder\_mead\_in\_selector,

 train\_option=train\_option,

 )

Библиотека размещена в репозитории: <https://github.com/aimclub/AutoTM>