ATC: Руководство пользователя

Версия 17.12.2020

***New**: Добавлена информация про запуск тестов и файлы с расшифровками кодов.*

Авторы: Константин Ломотин (), Екатерина Козлова ().

GitHub

Установка среды

1. Загрузить установочный файл для интерпретатора Python версии 3.7.7: https://www.python.org/ftp/python/3.7.7/python-3.7.7-amd64.exe
2. Установить интерпретатор.
3. Установить необходимые пакеты из файла ATC/requirements.txt (требуется доступ в интернет) командой `python -m pip install -r requirements.txt`
4. *[При необходимости]* Добавить папку с установленным интерпретатором в системную переменную `Path`.

[Примечание] Если при установке возникает ошибка доступа, можно попробовать добавить ключ `--user`:

`python -m pip install --user -r requirements.txt`

Запуск тестов [!]

После установки программы и моделей нужно запустить классификацию тестовых файлов. Этот шаг необходим, т.к. некоторые пакеты, такие как Pymystem3 и NLTK, могут загружать словари во время первого запуска. Также это позволит убедиться, что установка ATC прошла успешно.

Для запуска тестов нужно перейти в папку `ATC_v1.7.1/test/` и выполнить команду

```
python run_tests.py
```

Затем нужно убедиться, что все тесты завершились успешно (ожидается два сообщения “Не удалось распознать язык” для проверки пустых входных файлов). Результаты классификации лежат в `ATC_v1.7.1/test/out/`.

Запуск в графическом режиме

АТС можно запустить без аргументов, чтобы использовать графический интерфейс:

```
python ATC.py
```

В открывшемся окне можно ввести текст, открыть файл в одном из поддерживаемых форматов, настроить параметры классификации, посмотреть расшифровку кодов, экспортировать результат и посмотреть информацию о моделях.

Расшифровки доступны для рубрикаторов `SUBJ` и `IPV`. Файлы расшифровок загружаются из одноименных файлов в `ATC_v1.7.1/ATC/resources/`.

Запуск классификатора в пакетном режиме

Аргументы командной строки:

Аргумент	**Ключ**	**Полный ключ**	**Допустимые значения**	**Обязательное поле**
Входной файл	-i	--input	Путь к существующему файлу	да
Файл для сохранение результатов	-o	--output	Путь к файлу в существующей папке	да
Идентификатор рубрикатора	-id	--rubricator-id	Любой представленный в файле конфигурации модуля классификатора	да
Формат файла	-f	--format	plain, divided, multidoc, auto	да
Язык	-l	--language	auto, ru, en	да
Минимальная вероятность рубрики	-t	--threshold	Число от 0 до 1, десятичный разделитель - точка	нет

Аргумент	Ключ	Полный ключ	Допустимые значения	Обязательное поле
Нормализация	-n	--normalize	not, some, all	нет

- При значениях **auto** АТС попытается самостоятельно распознать язык и формат. Если это не получится, возникнет соответствующее сообщение об ошибке.
    - Для входного формата **Multidoc** необходимо, чтобы все тексты были на одном языке, т.к. язык автоматически определяется для всего файла.
- Без нормализации вероятность принадлежности текста каждому классу находится в диапазоне от 0 до 1 без ограничений, их сумма может превосходить 1. При нормализации вероятности масштабируются так, чтобы их сумма была равна 1. Нормализация выполняется делением на сумму вероятностей.
    - **not**: не выполнять нормализацию;
    - **some**: выполнять нормализацию, если сумма предсказанных вероятностей рубрик больше 1.
    - **all**: выполнить нормализацию.
- Рубрики с вероятностями ниже порога не попадут в результат.

Пример запуска:

```
python ATC.py -i ../data.csv -o ../result.csv -id SUBJ -f auto -l auto -t 0.75 -n some
```

Формат входных данных

Классификатор принимает тексты в одном из трех текстовых форматов: **plain**, **divided** и **multidoc**. Кодировка входного файла - `cp1251`.

Plain

Простой текст без разделения на поля. Может содержать произвольное количество строк. Пример:

> *Взрослые особи в среднем весят 30 грамм и имеют длину от 19 до 20 см. Спина тускло-серо-коричневая с характерной чёрной полосой под глазами от клюва до шеи[5]. Хвостовая часть бледно-белая, а грудь светло-лососевого цвета. Ноги и лапы чёрные, глаза коричневые, клюв чёрный, хвост серый, снизу кремовый с коричневатыми полосами. Молодые особи имеют более тусклый окрас с более коричневой полосой на глазах[3].*

Divided

Одна строка из трех компонентов, разделенных табуляцией: название, текст, ключевые слова. Табуляция обозначена стрелкой: →

Пример:

> *Mechanical analysis of functionally graded carbon nanotube reinforced composites: A review → Research activities related to functionally graded materials (FGMs) have increased rapidly in recent years. The superlative properties of carbon nanotubes, i.e. high strength, high stiffness, high aspect ratio and low density have made them an excellent reinforcement for composite materials. The present review includes: (i) a brief introduction of carbon nanotube reinforced composite (CNTRC) material; (ii) a review of mechanical analysis of FG-CNTRC; and (iii) a detailed discussion on the recent advances of FG-CNTRC and its prospect. → Functionally graded carbon nanotube reinforced composite; Mechanical analysis; Material.*

Multidoc

Файл с заголовком произвольным количеством строк. Аналогичен формату CSV с табулятором в качестве разделителя. Колонки:

- **id**: идентификатор текста;
- **title**: название;
- **body**: основной текст;
- **keywords**: ключевые слова;
- **correct**: содержит только признак валидности - три решетки `###`.

Пример:

> *id → title → body → keywords → correct*

> ***B11864306371** → Первичные материалы о коллемболах (Insecta: Collembola) пояса полярной пустыни Хибин → Большинство … проблем промышленной экологии Севера Кольского НЦ РАН. → Collembola\Мурманская область\Хибинские горы\биотопы\видовое разнообразие\новые находки\фауна → ###*

> ***B11864306428*** *→ Сравнительная характеристика зараженности промысловых пелагических и донных рыб Баренцева моря опасными для здоровья человека личинками нематоды Anisakis simplex → Сравнительный анализ показал существенные различия в характере зараженности личинками Anisakis simplex рыб, отличающихся по своей экологии и трофо-паразитарным связям. Россия, Полярный НИИ морскогорыбного хозяйства и океанографии им. Н.М.Книповича. → Anisakis simplex (Nem.)\Баренцево море\видовое разнообразие\новыенаходки\паразитофауна\промысловые рыбы → ###*

> ***B11864306754*** *→ Структура населения шмелей (Hymenoptera, Apidae, Bombus spp.) Восточной Фенноскандии → Установлено, что при изменении условий обитания шмели способны к трансформации жизненных циклов. Такая способность обусловлена как географической широтой, так и фенологическими и генетическими особенностями разных видов шмелей. Россия, Институт экологических проблем Севера УрО РАН. → Скандинавия\абиотические факторы\биотопы\видовое разнообразие\распределение по территории\фауна\шмели → ###*

Ключевые слова

В зависимости от языка ключевые слова имеют разные разделители.

- Для русского (код **ru**) - обратный слэш "\".
- Для английского (код **en**) - точка с запятой "**;**".

Формат сохранения результата

Результат классификации сохраняется в виде CSV-таблицы с табулятором в качестве разделителя. Выходной файл имеет кодировку `cp1251`. Таблица имеет разный вид в зависимости от формата входных текстов .

Результат для Plain и Divided

В начале файла записывается комментарий-заголовок, начинающийся с `#`. Он содержит информацию о запуске классификатора. Далее следуют строки с результатами классификации.

- Рубрики, вероятности которых меньше порога, не попадают в таблицу.
- По умолчанию порог равен нулю.
- Вероятности округляются до 5 знаков после десятичной точки.

- Первые три значения через точку в поле **Версия** показывают версию АТС, которая создала файл. Остальная часть версии устарела и не поддерживается.
- Если установлен слишком высокий порог вероятности, и в результат не попала ни одна рубрика, после заголовка ничего не идет.
- Случай, когда элементы вектора оказались меньше порога reject_threshold, заданного в файле конфигурации векторайзера, считается отказом от классификации. После заголовка в таком случае записывается значение **REJECT**.

> *# → ID рубрикатора → Язык → Порог вероятности → Версия приложения → Опция нормализации*

Затем

> *Код1 → Вероятность1*

> *Код2 → Вероятность2*

> *Код3 → Вероятность3*

> ...

Или

> *REJECT*

Пример, созданный АТС v1.7.0:

> *# → SUBJ → en → 0.3 → 1.7.0p3.0v3.0c → all*

> *f7 → 0.62908*

> *e9 → 0.16462*

> *f1 → 0.10469*

> *f2 → 0.03945*

Результат для Multidoc

Первая строка - заголовок таблицы, без # в начале:

> *id → result → rubricator → language → threshold → version → normalize → correct*

Далее в каждой строке идут значения этих полей, разделенные табуляцией.
Результат записывается в формате

> *Код1-Вероятность1/Код2-Вероятность2/Код3-Вероятность3/...*

- Если из-за высокого порога вероятности в результат не попала ни одна рубрика, в это поле ставится значение **EMPTY**.
- Если произошел отказ от классификации, в поле **result** записывается значение **REJECT**.

Пример:

> *id → result → rubricator → language → threshold → version → normalize → correct*

> ***B150720803779*** *→ f7-0.3119\f3-0.23092\f5-0.18432\f9-0.11344\e5-0.04524\e4-0.04362\f8-0.04013\f1-0.011\e1-0.0061\e8-0.00458\e2-0.00253\e9-0.00195\e3-0.00169\f4-0.00139\e7-0.00078\f2-0.00041 → SUBJ → en → 0.3 → 1.7.0p3.0v3.0c → all → ###*

> ***J0805808035*** *→ EMPTY → SUBJ → en → 0.3 → 1.7.0p3.0v3.0c → all → ###*

> ***J08423714116*** *→ REJECT → SUBJ → en → 0.3 → 1.7.0p3.0v3.0c → all → ###*

Запуск в режиме сервера

АТС может работать в серверном режиме, принимая TCP-соединения на указанном порту:

```
python ATC.py server 4242
```

Порт - обязательный аргумент.

После установки соединения сервер работает в последовательном режиме “запрос-ответ”. Запрос представляет собой JSON-строку:

```
{
  "id": "идентификатор текста",
  "title": "заголовок текста",
  "body": "непосредственно текст",
  "keywords": ["список", "ключевых", "слов"],
  "rubricator": "одно из ipv, grnti или subj",
  "language": "одно из en, ru или auto"
}
```

В ответ сервер посылает строку с сообщением об ошибке, **REJECT** или JSON-строку:

```
{
  "код1": вероятность1,
  "код2": вероятность2,
  ...
  "кодN": вероятностьN
}
```

Настройка конфигурации и установка моделей

Векторайзер

Файл конфигурации векторизатора находится по пути `ATC/analyzer/modules/word_embedding/config.ini`.

Единственная секция `Settings` содержит пять опций:

- **ru**: название файла модели Word2Vec.
- **en**: название файла англоязычной модели Word2Vec.
- **reject_threshold_sum**: порог отказа для пулинга по сумме.
- **reject_threshold_mean**: порог отказа для пулинга по среднему.
- **reject_threshold_max**: порог отказа для пулинга по максимуму.

Тип пулинга для каждой модели классификатора загружается автоматически из метаданных модели. Оптимальные значения порогов можно подобрать эмпирически, начав с малых - примерно 0.5 - и медленно повышая.

Установка модели состоит из двух шагов:

1. Поместить файлы модели `.model` и `.npy` в `ATC/analyzer/modules/word_embedding/`.
2. Указать имя `.model`-файла в файле конфигурации.

Можно помещать их в подпапки, например:

```
[Settings]
ru = ru_models/russian_w2v.model
en = en_models/english_w2v.model
```

Можно задать любой путь к файлу модели, записанный относительно папки `word_embedding` (абсолютные пути не поддерживаются).

Классификатор

Файл конфигурации классификатора находится по пути `ATC/analyzer/modules/classifier/config.ini`. Он состоит из произвольного количества секций. Каждая секция описывает модель для одного рубрикатора, название секции должно начинаться с **Rubr**. Опции каждой секции:

- **code**: идентификатор рубрикатора. Это значение можеть быть указано при запуске в пакетном режиме для ключа **-id** и будет доступно в графическом режиме на панели опций.
- **ru_model**: относительный путь к файлу русскоязычной модели `.plk`.
- **en_model**: относительный путь к файлу англоязычной модели `.plk`.

Если в настройках эксперимента ECS указать в качестве рубрикатора `SUBJ` или `IPV`, никаких проверок проводиться не будет, и система сможет обучать модели на произвольных рубрикаторах. ATC может загружать и использовать эти рубрикаторы без ограничений.

[Примечание] Нельзя указывать `RGNTI` для обучения модели на произвольном рубрикаторе, т.к. система поддерживает биение ГРНТИ на уровни. В теории, этим можно воспользоваться для обучения моели на произвольном трехуровневом рубрикаторе, разделенном точками.

Аналогично конфигурации векторизатора, пути к файлам моделей указываются относительно папки `classifier`.

Обязательно должна быть задана опция **name** и хотя бы один путь к модели.

Секции с пустым полем **name** не загружаются приложением.

Установка модели состоит из двух шагов:

1. Поместить файл модели `.plk` в `ATC/analyzer/modules/classifier/`.
2. Указать имя `.plk`-файла в файле конфигурации для соответствующего рубрикатора и языка.

[Примечание]

Количество установленных моделей влияет на скорость запуска АТС, но не влияет на скорость классификации.

Пример:

```
[Rubr0]
code = SUBJ
ru_model = remote/clf_model_ru_subj_perceptron_mean200_10_10_2020.plk
en_model = remote/clf_model_en_subj_perceptron_mean200_16_10_2020.plk

[Rubr1]
code = IPV
ru_model = remote/clf_model_ru_ipv_perceptron_mean200_10_10_2020.plk
en_model = remote/clf_model_en_ipv_perceptron_mean200_16_10_2020.plk

[Rubr2]
code = RGNTI
ru_model = remote/clf_model_ru_rgnti_perceptron_mean200_11_10_2020.plk
en_model = remote/clf_model_en_rgnti_perceptron_mean200_17_10_2020.plk

[Rubr3]
code = custom
ru_model = remote/clf_model_ru_subj_perceptron_mean200_17_10_2020.plk
en_model =
```