

PROMT Custom AI

Руководство администратора

PROMT Custom AI

Руководство администратора

Никакая часть настоящего документа не может быть воспроизведена без письменного разрешения компании PROMT (ООО «ПРОМТ»).

© 2003–2025, ООО «ПРОМТ». Все права защищены.

Ревизия 3.0

Россия, 199155,

Санкт-Петербург, Уральская ул., д. 17, лит. Е, кор. 3.

E-mail: common@promt.ru

support@promt.ru

Internet: <https://www.promt.ru>

<https://www.translate.ru>

Телефон / факс: +7 812 655-0350

PROMT®, ПРОМТ® — зарегистрированные торговые марки ООО «ПРОМТ».

Все остальные торговые марки являются собственностью соответствующих владельцев.

Оглавление

1. О документации	5
1.1. Состав документации	5
1.2. Условные обозначения	5
2. Введение	6
2.1. Назначение PROMT Custom AI	6
2.2. Помощь и сопровождение	6
3. Установка	7
3.1. Системные требования	7
3.2. Установка продукта	9
3.2.1. Установка языковых пакетов	11
3.3. Активация	13
3.4. Настройки конфигурационного файла appsettings.json	14
3.4.1. Настройка доступа по https-протоколу	14
3.4.2. Задание индекса используемого GPU	15
3.4.3. Дополнительная настройка параметров работы PAI	16
3.4.4. Настройка почтовых уведомлений	17
3.4.5. Назначение папки для сохранения файлов при подсчете BLEU	19
3.4.6. Отключение проверки SSL	19
3.5. Веб-интерфейс PROMT Custom AI	21
3.6. Удаление PROMT Custom AI	24
4. Общая схема работы	25
5. Подготовка исходных данных	26
6. Обучение модели	28
6.1. Задание параметров обучения и запуск процесса обучения	28
6.2. Процесс обучения	33
6.2.1. Этапы обучения модели	34
6.3. Результат обучения	38
7. Загрузка настроенной модели на сервер	40
8. Оценка результатов	43
Приложение 1. Глоссарий	46
Приложение 2. Префиксы для обозначения языков	48
Приложение 3. Установка необходимых компонентов для Python на системах Linux	50
Ubuntu 20.04	50
Ubuntu 22.04	51
Ubuntu 24.04	52
Debian 11	53
Debian 12	54
Astra Linux 1.7 SE	55

Astra Linux 1.8 SE.....	56
ALT Linux 10.0.....	57
РЕД ОС 7.3.....	58
РЕД ОС 8	59

1. О документации

1.1. Состав документации

Документация PROMT Custom AI содержит сведения об установке указанного компонента, а также информацию по настройке обучения моделей нейронного машинного перевода (NMT-моделей), используемых в переводчике PROMT, с помощью корпуса параллельных текстов.

Данное руководство предназначено для пользователей PROMT Neural Translation Server (PNTS), включая его модификации, а также для пользователей PROMT Translation Factory (PTF), осуществляющих настройку NMT-моделей посредством дополнительной функциональности переводчика, реализованной в компоненте PROMT Custom AI.

 Подробные сведения о работе с PROMT Neural Translation Server см. в руководстве пользователя и в руководстве администратора PNTS. Информацию о работе с PROMT Translation Factory вы найдете в руководстве пользователя и в руководстве администратора PTF.

1.2. Условные обозначения

В документации используются следующие условные обозначения:



— советы и рекомендации;



— важные замечания;



— примечания или дополнительная информация;



— ссылка на подробные сведения в руководстве.

2. Введение

В программных продуктах PROMT используется нейронный машинный перевод (*NMT, от англ. Neural Machine Translation*). NMT – это актуальная технология машинного перевода, которая использует нейронные модели, обученные на параллельных корпусах. Нейронный перевод требует для обучения параллельные корпуса, позволяющие сравнить автоматический перевод с эталонным «человеческим», при этом в процессе обучения он оперирует не отдельными фразами и словосочетаниями, а целыми предложениями.

2.1. Назначение PROMT Custom AI

PROMT Custom AI (сокращенно PAI) является дополнением для программных продуктов PROMT Neural Translation Server (далее PNTS), включая его модификации, и PROMT Translation Factory (далее PTF). PROMT Custom AI позволяет расширить функциональность программных продуктов PNTS и PTF, установленных у пользователя, за счет дополнительной функциональной возможности автоматической настройки (обучения) моделей нейронного машинного перевода.

Входными данными для настройки являются NMT-модель (в виде zip-архива) и корпус параллельных текстов, предоставляемый пользователем*.

* На англо-русском и русско-английском направлениях для обучения моделей могут использоваться данные противоположного языкового направления.

Результатом работы является настроенная по предоставленному текстовому корпусу новая модель (в виде zip-архива). Обученную модель необходимо загрузить в PNTS или PTF.

2.2. Помощь и сопровождение

В случае возникновения проблем при работе с PROMT Custom AI следует обратиться в отдел технической поддержки по телефону или отправить сообщение по электронной почте. При этом укажите следующее:

- языковые пары, которые используются с данным программным продуктом;
- основные характеристики компьютера: тип процессора, объем оперативной памяти, объем свободного места на жестком диске, наличие сети, GPU, версия драйвера видеокарты и версия CUDA;
- характеристики используемого ПО: имя и версия дистрибутива Linux (например, Ubuntu 20.04.1 LTS), имя и версия ядра Linux (результат вызова команды `uname -a`);
- суть проблемы и действия, предшествовавшие ее появлению;
- действия, предпринятые для решения данной проблемы;
- при получении сообщения об ошибке – его точный текст или снимок экрана с этим сообщением.

При обращении в отдел технической поддержки по телефону рекомендуется находиться перед компьютером.

3. Установка

PROMT Custom AI устанавливается с помощью инсталляционного набора.

 Установка должна осуществляться пользователем с правами root.

3.1. Системные требования

Для работы PROMT Custom AI потребуется компьютер со следующими характеристиками:

- Процессор класса Intel Core i5 (или выше) или Xeon E3 (или выше) с 4 ядрами минимум (рекомендуется 8 ядер)
- Графический процессор (GPU) с выделенной видеопамятью не менее 16 Гб, с поддержкой технологии CUDA 12 и CUDA Compute Capability не ниже 7.5*
* требуется установить последнюю версию драйвера GPU для соответствующей ОС. Чем больший объем памяти будет выделен, тем лучше будет результат обучения модели. Значение характеристики CUDA Compute Capability для конкретной модели GPU указано на официальном сайте производителя GPU.
- Оперативная память: 32 Гб (при объеме тренировочных данных до 1 млн. сегментов)
- Место на диске: 50 Гб

Поддерживается работа на следующих ОС*:

- Ubuntu 20.04
- Ubuntu 22.04
- Ubuntu 24.04
- AstraLinux SE 1.7
- AstraLinux SE 1.8
- Debian 11
- Debian 12
- РЕД ОС 7.3
- РЕД ОС 8
- ALT Linux 10.0

* Рекомендуется установка всех доступных обновлений для операционной системы.

Целевые ОС должны включать следующие системные библиотеки и компоненты:

- GLIBC версии 2.17 или выше

- libgcc_s.so.1
- libstdc++.so.6
- менеджер системных служб systemctl.

Дополнительно перед установкой продукта требуется установка следующих пакетов (дистрибутивы входят в поставку):

- Python версии от 3.6.4 до 3.13 включительно
- пакет Numpy для Python
- пакет PyYAML <7, ≥5.3 для Python*
- пакет CTranslate2 для Python, версия 2.14 или 2.23 (в зависимости от версии Python).

*В комплект поставки входит пакет PyYAML 6.0.

Установка необходимых пакетов выполняется из папки Prerequisites/python3 распакованного установочного набора (подробнее см. в [Приложении 3](#)).

Программное обеспечение топо, которое используется при работе PROMT Custom AI, также входит в состав дистрибутива.

Веб-интерфейс продукта поддерживает работу в последних версиях следующих браузеров:

- Chrome
- Firefox
- Opera
- Edge
- Яндекс-браузер.

3.2. Установка продукта

PROMT Custom AI (PAI) функционирует совместно с программными продуктами PNTS и PTF. Для использования моделей, обученных с помощью PAI, требуется наличие у пользователя PNTS или PTF с действующей лицензией.

Перед началом установки необходимо установить команды `less` и `acl`:

```
sudo apt update
sudo apt install less
sudo apt install acl
```

Распакуйте архив, переданный вам при покупке программы.

Например:

```
sudo unzip ./PAI1.1_PR24_0115.zip
```

Результатом выполнения команды будет установочный пакет, распакованный в текущую папку.

Папка с распакованным архивом содержит установочный `run`-файл и папку с пререквизитами.



Перед установкой PAI требуется установка Python и необходимых пакетов.

Информацию об установке необходимых пререквизитов вы найдете в [Приложении 3](#).

Для установки PROMT Custom AI запустите установочный файл `setup.run`:

```
sudo chmod +x setup.run && sudo ./setup.run
```

Процесс установки состоит из нескольких шагов, которые выполняются автоматически.

1. Ознакомьтесь с лицензионным соглашением и примите его условия.



При отказе от условий лицензионного соглашения дальнейшая установка невозможна.

2. Укажите пользователя сервисов PAI.

По умолчанию пользователь – `rai`.

3. Укажите версию Python.

По умолчанию – `python3`



Следует указывать именно ту версию Python, с которой предстоит работать.



При установке набора на Ubuntu 24.04, Debian 12 и Astra Linux 1.8 SE необходимо указывать `python3` из виртуального окружения `/path/to/venv/bin/python3`, в противном случае установленные пакеты находиться не будут. Подробнее об установке пререквизитов для Python на различных системах Linux см. в [Приложении 3](#).

 В случае отсутствия Python или пререквизитов, установка прекратится и в консоль будет выведено соответствующее сообщение.

4. Укажите папку, в которую будет установлен программный продукт.

По умолчанию продукт устанавливается в папку `/usr/local/paiXX` (XX – номер версии продукта).

Например: `/usr/local/pai1.1`

5. Программные модули распаковываются в корневую папку продукта. Происходит настройка исполняемых файлов и установка разрешений.

6. Укажите путь к папке `storage`.

Путь к папке по умолчанию: `/usr/local/paiXX/addon/storage` (где XX – номер версии продукта).

Например: `/usr/local/pai1.1/addon/storage`

В случае указания пользовательского пути к папке `storage` необходимо дать права пользователям на папку `storage`.

 В папке `storage` будут храниться файлы, генерируемые в процессе работы PROMT Custom AI. У пользователя `pai` должны быть полные права на каталоги, в которых располагается папка `storage`.

7. Укажите адрес, по которому расположен веб-интерфейс PNTS или PTF.

 Адрес PNTS при установке по умолчанию `http://127.0.0.1/pnts`. Адрес PTF при установке по умолчанию `http://127.0.0.1/ptf`.

 Заменить адрес сервера можно только в файле конфигурации web-приложения `appsettings.json` (по умолчанию `/usr/local/pai1.1/addon/app/appsettings.json`) в секциях `"BleuUtility"` и `"PntsProxy"`. После уточнения файла конфигурации необходимо перезапустить сервис `pai` (см. [Активация](#)).

8. Настройка фаервола.

По умолчанию порт для входящих соединений – 80. Если порт 80 занят, укажите другой номер порта.

9. Задайте имя веб-приложения.

По умолчанию имя веб-приложения – `pai`.

10. Инсталлятор производит распаковку и установку приложения, настройки модифицируются под текущее окружение ОС.

11. Инсталлятор предложит выполнить установку языковых пакетов.

При утвердительном ответе инсталлятор запрашивает точный путь к архиву с пакетами. Архив представляет собой файл .zip. В качестве пути принимаются путь к папке с пакетами, путь к пакету, а также перечисление путей к папкам и файлам через пробел.

✿ Языковые пакеты входят в состав набора. Каждый пакет представляет собой архив, в который входит NMT-модель и корпус общелексических данных. Модели из установленных языковых пакетов будут отображаться на странице *Построение модели* в поле *Базовая модель*.

В случае отказа вы можете установить языковые пакеты позже. Для этого следует использовать скрипт установки пакетов `install_models.sh`, располагающийся в каталоге `scripts` установленного набора (по умолчанию `/usr/local/pai1.1/addon/scripts/install_models.sh`).

Подробнее об установке языковых пакетов см. в разделе [Установка языковых пакетов](#).

Сразу после установки на операционные системы, где используется модуль безопасности ядра SELinux, на сервере может появиться уведомление безопасности SELinux:

1. SELinux запрещает `/usr/lib/systemd/systemd` доступ `execute` к файл `dotnet`

Разрешите доступ, создав локальный модуль политики:

```
sudo ausearch -c '(dotnet)' --raw | audit2allow -M my-dotnet
sudo semodule -X 300 -i my-dotnet.pp:
```

2. SELinux запрещает `/usr/lib/systemd/systemd` доступ `execute_no_trans` к файл `/usr/local/pai1.1/addon/dotnet/dotnet`

Измените метку каталога с помощью следующей команды:

```
sudo /sbin/restorecon -v /usr/local/pai1.1/addon/dotnet/dotnet
```

⚠ В случае игнорирования данных уведомлений безопасности SELinux при установке на системы с SELinux веб-интерфейс PROMT Custom AI будет недоступен.

3.2.1. Установка языковых пакетов

Если языковые пакеты не были установлены при установке PAI, они могут быть установлены позже с помощью скрипта `install_models.sh`, который находится в каталоге `scripts` установленного набора (по умолчанию `/usr/local/pai1.1/addon/scripts/install_models.sh`).

✿ Языковые пакеты входят в состав набора. Каждый пакет представляет собой архив с расширением .zip, в который входит NMT-модель и корпус общелексических данных. Модели из установленных языковых пакетов будут отображаться на странице *Построение модели* в поле *Базовая модель*.

Название архива должно соответствовать следующему шаблону:
[название пакета]_{направление}_[доп. информация].zip

[название пакета] может содержать только буквы латинского алфавита, цифры, дефисы и нижнее подчеркивание. Если название пакета при запуске скрипта не указано, будет использоваться название basic.

{направление} – это обязательная часть названия языкового пакета. В качестве указателей языкового направления можно использовать внутренние префиксы PROMT, указываемые в названиях NMT-моделей (например, ge, [ar]r и т.п.) или RFC-префиксы (например, [en][ru], [lt][ru] и т.п.).

 Таблицу языковых префиксов в формате RFC вы найдете в [Приложении 2](#).

[доп. информация] – опциональная часть.

Перед началом установки пакетов перейдите в каталог scripts установленного набора.

Убедитесь, что у пользователя, выполняющего установку пакетов, есть права на чтение из папки с пакетами.

- Для установки всех пакетов из папки запустите скрипт с точным указанием пути к папке с пакетами

```
sudo ./install_models.sh <путь к папке с пакетами>
```

Например, если пакеты лежат по пути /usr/local/models, то для установки всех имеющихся моделей из папки models нужно ввести следующую команду:

```
sudo ./install_models.sh /usr/local/models
```

- Для установки конкретного пакета запустите скрипт с точным указанием пути к архиву с пакетом

```
sudo ./install_models.sh <путь к архиву пакета>
```

Например, если пакеты лежат по пути /usr/local/models, то для установки пакета pack_[en][ru].zip следует ввести команду

```
sudo ./installmodels.sh /usr/local/models/pack_[en][ru].zip
```

- Для установки двух пакетов запустите скрипт с точным указанием пути к каждому пакету через пробел

```
sudo ./install_models.sh <путь к архиву пакета 1> <путь к архиву пакета 2>
```

Например,

```
sudo ./install_models.sh /usr/local/models/pack_[en][ru].zip  
/usr/local/models/pack_[lt][ru].zip
```

3.3. Активация

Доступ к основной функциональности PROMT Custom AI определяется параметрами действующего файла лицензии.

Лицензионный файл `license.lic` устанавливается вместе с набором. Файл находится в корневой папке набора (по умолчанию `/usr/local/paiXX`, где XX – номер версии).

Работа PROMT Custom AI невозможна без активации.

Чтобы активировать продукт, выполните следующее:

1. Получите идентификатор компьютера, привязанный к оборудованию системы, в которой установлен продукт.

Для этого запустите скрипт `hwid.sh`, который находится в папке с установленным продуктом (по умолчанию `/usr/local/pai1.1/hwid.sh`):

```
sudo ./hwid.sh
```

В результате выполнения скрипта в консоль будет выведена строка с идентификатором компьютера.

2. Пришлите в службу поддержки компании ПРОМТ лицензионный файл и идентификатор компьютера (hardware id).
3. Получив новый лицензионный файл, скопируйте его "поверх" файла, установленного с набором.
4. Назначьте пользователя `pai:pai` владельцем файла лицензии и дайте права на чтение файла всем группам пользователей:

Для этого выполните команды

```
sudo chmod +r /usr/local/pai1.1/license.lic && sudo chown pai:pai /usr/local/pai1.1/license.lic
```

✿ В приведенном выше примере указан путь при установке в папку по умолчанию.

Перезапустите сервис `pai`:

```
sudo systemctl restart promt.customai.service
```

Результатом активации является новый лицензионный файл, который привязан к оборудованию. Файл содержит информацию о доступных языковых парах и позволяет работать с веб-интерфейсом PROMT Custom AI.

3.4. Настройки конфигурационного файла appsettings.json

 После уточнения файла конфигурации обязательно перезапустите сервис pai (см. раздел [Активация](#)).

3.4.1. Настройка доступа по https-протоколу

По умолчанию PROMT Custom AI устанавливается с доступом только по http-протоколу.

Администратор может настроить доступ по https-протоколу. Для этого необходимо выполнить следующее:

1. Добавьте блок *"Https"* в конфигурационный файл appsettings.json (по умолчанию /usr/local/pai1.1/addon/app/appsettings.json)

2. Пропишите в добавленном блоке следующие параметры:

<i>Url</i>	url, на котором будет работать PAI
<i>Certificate</i>	данные сертификата
<i>Path</i>	путь к сертификату
<i>KeyPath</i>	путь к ключу
<i>Password</i>	пароль к сертификату*
<i>AllowInvalid</i>	Разрешение на использование самоподписанных сертификатов. Значение по умолчанию – "true".

*Если у ключа нет пароля, то строку *"Password"* в файл конфигурации добавлять не нужно.

 Ключ и сертификат должны быть совместимы с настроенной и поддерживаемой версией TLS на сервере. Минимальная поддерживаемая версия протокола TLS – 1.2.

3. Положите используемые ключ и сертификат в папку набора (по умолчанию /usr/local/pai1.1).

4. Выдайте права на чтение сертификата и ключа всем группам пользователей.

```
sudo chmod a+r /path/to/certificate.pem
sudo chmod a+r /path/to/key.key
```

Пример:

```
sudo chmod a+r /usr/local/pai1.1/certificate.pem
sudo chmod a+r /usr/local/pai1.1/key.key
```

Пример секции файла appsettings.json до настройки:

```
"Kestrel": {
  "Endpoints": {
    "Http": {
      "Url": "http://+:81"
    }
  }
},
```

Пример настройки:

```
"Kestrel": {
  "Endpoints": {
    "Http": {
      "Url": "http://+:81"
    },
    "Https": {
      "Url": "https://+:443",
      "Certificate": {
        "Path": "/usr/local/pai1.1/certificate.pem",
        "KeyPath": "/usr/local/pai1.1/key.key",
        "Password": "qwe123",
        "AllowInvalid": true
      }
    }
  }
},
```

 Перед настройкой доступа по https следует убедиться, что нужный порт не занят. Если в системе больше никакая программа не работает на https, то можно использовать порт 443, в противном случае выбирайте следующий свободный порт (444 и т.д.)

3.4.2. Задание индекса используемого GPU

В том случае если предполагается работать на системе с несколькими GPU, чтобы выбрать определенное устройство, необходимо прописать его индекс в параметре *"Model_GPU_Devices"* конфигурационного файла *appsettings.json* (по умолчанию */usr/local/pai1.1/addon/app/appsettings.json*).

Все GPU-устройства, подключенные к компьютеру, получают числовые идентификаторы от 0 до N-1, где N – число GPU-устройств на машине. Для определения индекса GPU вызовите утилиту *promt-gpuinfo.exe*, которая находится в папке *utils* установленного продукта.

Пример вызова утилиты при установке в папку по умолчанию:

```
/usr/local/pai1.1/utils/promt-gpuinfo.exe
```

Первое выведенное в результатах устройство имеет индекс "0".

По умолчанию в параметре *"Model_GPU_Devices"* указано одно устройство GPU с индексом "0". Для корректной работы на нескольких устройствах GPU необходимо указать используемые GPU в параметре *"Model_GPU_Devices"* через пробел:

Пример настройки для работы на двух GPU с индексами "0" и "1":

```
"TrainingUtility":  
{  
  ...  
  "Model_GPU_Devices": "0 1"  
  ...  
}
```

Если любой из указанных идентификаторов выходит за границы диапазона, процесс обучения будет остановлен с сообщением об ошибке.

 Все устройства GPU, указанные в параметре *"Model_GPU_Devices"*, должны соответствовать минимальным требованиям (см. раздел [Системные требования](#)).

3.4.3. Дополнительная настройка параметров работы PAI

Основная настройка перед запуском обучения осуществляется через веб-интерфейс PROMT Custom AI на странице *Данные для обучения* группы страниц *Построение модели* (подробнее см. [Задание параметров и запуск процесса настройки](#)). В случае необходимости вы можете дополнительно уточнить некоторые параметры работы PAI в конфигурационном файле *appsettings.json* (по умолчанию */usr/local/pai1.1/addon/app/appsettings.json*). Параметры могут быть настроены в секции *"DefaultSettings"* раздела *"TrainingUtility"*.

Доступна настройка следующих параметров:

<i>Training_Set_Min_Size</i>	Нижняя граница количества входных данных для обучения: минимальный размер, до которого дублируется пользовательский корпус. По умолчанию – 1000000 строк.
<i>Training_Iteration_Count</i>	Количество итераций* обучения. Значение по умолчанию – 1000.
<i>Max_Training_Duration_Hours</i>	Максимальная длительность обучения в часах. Значение по умолчанию – 48.
<i>Early_Stopping</i>	Количество итераций* без улучшений, которое выполнит <i>maian</i> , прежде чем остановить обучение. Значение по умолчанию – 10.
<i>Keep_Best</i>	Сохраняемая модель: <i>"true"</i> – по окончании обучения сохраняется лучшая модель, <i>"false"</i> – сохраняется последняя полученная модель. Значение по умолчанию – <i>"true"</i> .

python_path

Путь для вызова python нужной версии, настраивается при установке продукта. По умолчанию – python3. Версия Python должна соответствовать системным требованиям и для нее должны быть установлены соответствующие пакеты (см. раздел 3.1)**.

CT2

Автоматическая конвертация тренируемой модели marian в CT2.
Если конвертация в CT2 необходима, укажите значение "1".
Если для данного параметра указано значение "0", конвертация производиться не будет. Значение по умолчанию – 1 (конвертация включена).

* Итерация – это полный проход по всему корпусу обучающих данных за один цикл обучения модели.

** На Ubuntu 24.04, Debian 12 и Astra Linux 1.8 SE следует указывать python3 из виртуального окружения /path/to/venv/bin/python3. Подробнее об установке пререквизитов для Python на различных системах Linux см. в [Приложении 3](#).

✿ Если после завершения очередной итерации время работы программы превысило время, выделенное на обучение, цикл обучения прервется на том результате, который получен в последней итерации, и полученная модель будет конвертирована в CT2 (по умолчанию) и сохранена.

Пример настроек:

```
"DefaultSettings": {  
...  
  "Training_Set_Min_Size": "1000000",  
  "Python_Path": "python3",  
  "CT2": "1",  
  "Training_Iteration_Count": "1000",  
  "Max_Training_Duration_Hours": "48",  
  "Early_Stopping": "10",  
  "Keep_Best": "true"  
},  
...
```

3.4.4. Настройка почтовых уведомлений

Чтобы получать email-оповещения по окончании обучения модели в PROMT Custom AI, настройте почтовые уведомления.

✿ По умолчанию почтовые уведомления отключены.

Чтобы включить почтовые уведомления, необходимо внести уточнения в секцию *"MailNotify"* конфигурационного файла *appsettings.json* (по умолчанию */usr/local/pai1.1/addon/app/appsettings.json*).

Секция *"MailNotify"* по умолчанию имеет следующий вид:

```
"MailNotify": {  
  "Host": "",  
  "Port": 25,  
  "From": "",  
  "UserName": "",  
  "Password": "",  
  "Users": [],  
  "TemplateLanguage": "ru"  
}
```

Внесите необходимые уточнения в следующие параметры:

<i>Host</i>	Адрес почтового сервера. Если адрес не задан, почтовые уведомления отключены.
<i>Port</i>	Порт для подключения к почтовому серверу. По умолчанию номер порта 25. Если сервер поддерживает защищенное подключение (SSL или TLS), то номер порта – 465.
<i>From</i>	Имя отправителя, от которого будут приходить уведомления.
<i>UserName</i>	Данные для подключения к почтовому серверу.
<i>Password</i>	Пароль для подключения к почтовому серверу.
<i>Users</i>	Список пользователей, которым будет отправляться почтовое уведомление. Адреса почты следует указывать в кавычках через запятую, например: <i>"Users": ["username1@company.ru", "username2@company.ru"]</i>
<i>TemplateLanguage</i>	Локализация почтовых уведомлений. Поддерживаются русская ("ru") и английская ("en") локализации. По умолчанию уведомления приходят на русском языке.

Почтовые уведомления получают пользователи, указанные в параметре *"Users"*.

Пример настроек:

```
"MailNotify": {  
  "Host": "mail.company.ru",  
  "Port": 25,  
  "From": "tb@company.ru",  
  "UserName": "company\\tb",  
  "Password": "jAtdYWGr,Lp$XR-Z",  
}
```

```

    "Users": ["username1@company.ru", "username2@company.ru"],
    "TemplateLanguage": "ru"
}

```

Предусмотрены следующие типы оповещений:

- Обучение завершено
- Обучение завершено с ошибкой
- Обучение остановлено пользователем*.

* Данное оповещение отправляется в том случае, если обучение было остановлено пользователем с помощью кнопки *Остановить обучение*.

3.4.5. Назначение папки для сохранения файлов при подсчете BLEU

Чтобы назначить папку, в которую будут сохраняться файлы при подсчете BLEU, укажите её название в параметре `"TranslationFilesDirectory"` секции `"Bleu"` конфигурационного файла `appsettings.json` (по умолчанию `/usr/local/pai1.1/addon/app/appsettings.json`).

Пример настройки для сохранения файлов в папку `scores`:

```

    "Files": {
    ...
    },
    "Bleu": {
    ...
    "TranslationFilesDirectory": "/scores"
    }

```

 Указанная в параметре `"TranslationFilesDirectory"` папка будет создана в папке `storage/bleu` (по умолчанию `/usr/local/pai1.1/addon/storage/bleu`).

3.4.6. Отключение проверки SSL

При входе в веб-интерфейс *PAI* по умолчанию происходит обращение к серверу *PNTS* для получения информации об имеющихся языковых направлениях. При этом в случае возникновения ошибки проверки сертификата SSL, выводится сообщение об ошибке.

Чтобы отключить проверку сертификата ssl при обращении к серверу *PNTS*, добавьте параметр `"SslCertificateValidation"` со значением `"false"` в конец основной группы настроек конфигурационного файла `appsettings.json` (по умолчанию `/usr/local/pai1.1/addon/app/appsettings.json`).

Пример настройки:

```

...
"MailNotify": {
  "Host": "",
  "Port": 25,
  "From": "",
  "UserName": "",
  "Password": "",

```

```
"Users": [],  
"TemplateLanguage": "ru"  
},  
"SslCertificateValidation": false  
}
```

❗ После внесения изменений в файл конфигурации перезапустите сервис приложения.

✿ Отключить проверку сертификата ssl можно при первом входе в веб-интерфейс (см. [Веб-интерфейс PROMT Custom AI](#)).

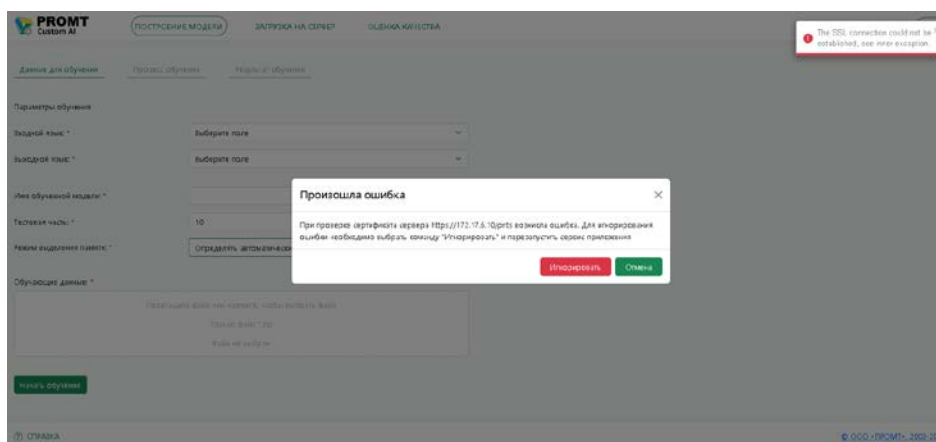
3.5. Веб-интерфейс PROMT Custom AI

Для доступа к веб-интерфейсу PROMT Custom AI запустите браузер и введите адрес [http\[s\]://<имя сервера или IP-адрес>/<имя веб-приложения>](http[s]://<имя сервера или IP-адрес>/<имя веб-приложения>).

✿ Имя веб-приложения по умолчанию – `rai`.

✿ По умолчанию *PAI* устанавливается с доступом только по `http`-протоколу. Доступ по `https`-протоколу требует дополнительной настройки (см. [Настройка доступа по https-протоколу](#)).

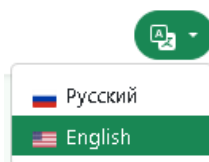
При входе в веб-интерфейс *PAI* по умолчанию происходит обращение к серверу *PNTS* для получения информации об имеющихся языковых направлениях. Если при обращении к серверу произошла ошибка проверки сертификата SSL, вы увидите сообщение об ошибке в правом верхнем углу страницы и диалоговое окно с предложением отключить проверку сертификата:



Чтобы отключить проверку сертификата `ssl` при обращении к серверу *PNTS*, нажмите на кнопку *Игнорировать* и перезапустите сервис приложения.

✿ Локализация веб-интерфейса *PAI* при первом входе зависит от настроек браузера. Если в браузере выбран русский язык, интерфейс в *PAI* будет на русском. Во всех остальных случаях интерфейс будет на английском.

Чтобы изменить язык веб-интерфейса, нажмите на иконку  в правом верхнем углу страницы и выберите язык



Приложение открывается на странице *Построение модели*:

ПROMT Custom AI

ПОСТРОЕНИЕ МОДЕЛИ ЗАГРУЗКА НА СЕРВЕР ОЦЕНКА КАЧЕСТВА

Данные для обучения | Процесс обучения | Результат обучения

Параметры обучения

Входной язык: * Выберите поле

Выходной язык: * Выберите поле

Имя обученной модели: * ZIP

Тестовая часть: * 10 %

Режим выделения памяти: * ☒ Определять автоматически

Обучающие данные: *

Перетащите файл или нажмите, чтобы выбрать файл

Только файл *.zip

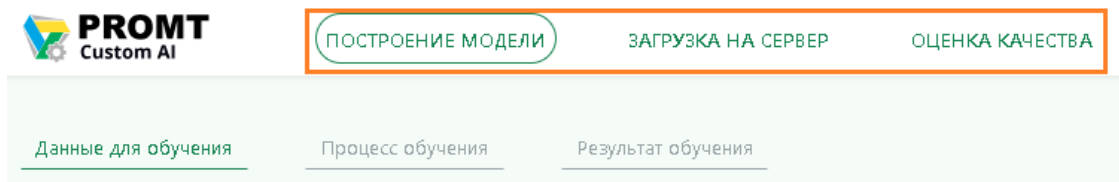
Файл не выбран

Начать обучение

СПРАВКА © ООО «ПРОМТ», 2009-2025

Рис. 3.1 Страница *Построение модели*

Главное меню PROMT Custom AI в верхней части страницы представлено в виде следующих заголовков:



• *Построение модели*

Страница *Построение модели* содержит несколько вкладок, обеспечивающих доступ к следующей функциональности:

- *Данные для обучения* – настройка параметров обучения и запуск обучения.
- *Процесс обучения* – контроль процесса обучения модели.
- *Результат обучения* – скачивание обученной модели/ текстовой части/ лога обучения и переход на стартовую страницу для новой сессии обучения.



Переход между вкладками при обучении модели осуществляется автоматически. После запуска обучения открывается вкладка *Процесс обучения*, по окончании обучения открывается вкладка *Результат обучения*.

• *Загрузка на сервер*

Страница *Загрузка на сервер* позволяет загрузить настроенную модель на сервер PNTS или PTF.

- **Оценка качества**

Страница *Оценка качества* позволяет выполнить оценку результатов настройки модели.

В нижнем левом углу страницы расположена кнопка  *Справка*, по которой доступно скачивание документации к продукту.

3.6. Удаление PROMT Custom AI

Чтобы удалить продукт, запустите скрипт `uninstall.sh` из папки `uninstall` в корневой папке продукта (по умолчанию `/usr/local/pai1.1/addon/uninstall/uninstall.sh`):

```
sudo ./uninstall.sh
```

При запуске скрипт проверит, запущен ли он с правами суперпользователя, и запросит у пользователя явное разрешение на удаление продукта.

 Скрипт полностью удаляет PAI из системы, в том числе системный сервис и корневую папку продукта. Созданный при установке продукта пользователь и правила для Firewall (UFW) не удаляются. Если для папки `storage` был указан пользовательский путь, то при удалении PAI с помощью скрипта папка не будет удалена.

4. Общая схема работы

Общая схема работы может быть представлена в виде следующих шагов:

1. Подготовка исходных данных

Подготовьте корпус параллельных текстов, на базе которых планируется провести обучение NMT-модели.

 Подробнее о подготовке данных для обучения см. в разделе [Подготовка исходных данных](#).

2. Настройка модели

Задайте необходимые параметры на странице *Построение модели* и запустите обучение.

 Подробные сведения о запуске процесса настройки модели см. в разделе [Задание параметров и запуск процесса настройки](#).

 Некоторые параметры запуска утилиты обучения могут быть заданы в конфигурационном файле *appsettings.json* (см. [Настройки конфигурационного файла appsettings.json](#)).

3. Сохранение результатов настройки модели

Сохраните результаты на странице *Результат обучения* блока *Построение модели* (см. [Результат обучения](#)).

4. Загрузка настроенной модели на сервер

Установите настроенную NMT-модель в *PNTS* или *PTF* на странице *Загрузка на сервер* (см. раздел [Загрузка настроенной модели на сервер](#)).

5. Оценка результатов настройки модели с помощью значения BLEU

 Подробные сведения об оценке результатов настройки вы найдете в разделе [Оценка результатов](#).

5. Подготовка исходных данных

Исходные данные (текстовый корпус) должны быть представлены в виде архива формата zip. Содержимое архива – это один или несколько tmx-файлов и/или текстовые файлы в кодировке UTF-8.

Текстовые файлы должны иметь расширения, соответствующие префиксам входного или выходного языков.

 Название текстовых файлов может быть произвольным (за исключением списка зарезервированных имен файлов). Необходимо, чтобы в папке с тестовыми данными имелась пара файлов с одним и тем же названием, но имеющих разные расширения, соответствующие префиксам входного и выходного языков.

 Таблицу с языковыми префиксами вы найдете в [Приложении 2](#).

Минимальный рекомендуемый размер пользовательского корпуса – 5000 строк.

Список зарезервированных (служебных) имен файлов

Условные обозначения:

[\$src] – префикс входного языка

[\$tgt] – префикс выходного языка

aligned-grow-diag-final.realigned	corpus.bpe.[\$src]
corpus.bpe.[\$src].sub	corpus.bpe.[\$tgt]
corpus.bpe.[\$tgt].sub	corpus.[\$src]
corpus.[\$src].o	corpus.full.bpe.[\$src]
corpus.full.bpe.[\$tgt]	corpus.full.bpe.with.basic.[\$src]
corpus.full.bpe.with.basic.[\$tgt]	corpus.full.bpe.[\$src].dupl
corpus.full.bpe.[\$tgt].dupl	corpus.[\$tgt]
corpus.[\$tgt].o	corpus.tok.def.clean.[\$src]
corpus.tok.def.clean.[\$tgt]	corpus.tok.def.[\$src]
corpus.tok.def.[\$tgt]	corpus.tok.[\$src]
corpus.tok.[\$src].sub	corpus.tok.[\$tgt]
corpus.tok.[\$tgt].sub	aligned-grow-diag-final.realigned.dupl
general.[\$src].part	general.[\$tgt].part
general.tok.def.[\$src]	general.tok.def.[\$tgt]
general.tok.def.clean.[\$src]	general.tok.def.clean.[\$tgt]
general.tok.[\$src]	general.tok.[\$tgt]
general.bpe.[\$src]	general.bpe.[\$tgt]
dev.bpe.[\$src]	dev.bpe.[\$src].output
dev.bpe.[\$tgt]	dev.[\$src]
dev.[\$tgt]	dev.tok.[\$src]
dev.tok.[\$tgt]	test.[\$src]
test.[\$tgt]	

Таблица 5.1 Зарезервированные имена файлов

Примеры файлов для англо-русского направления:

aligned-grow-diag-final.realigned	corpus.bpe.en
corpus.bpe.en.sub	corpus.bpe.ru
corpus.bpe.ru.sub	corpus.en
corpus.en.o	corpus.full.bpe.en
corpus.full.bpe.ru	corpus.ru
corpus.ru.o	corpus.tok.def.clean.en
corpus.tok.def.clean.ru	corpus.tok.def.en
corpus.tok.def.ru	corpus.tok.en
corpus.tok.en.sub	corpus.tok.ru
corpus.tok.ru.sub	dev.bpe.en
dev.bpe.en.output	dev.bpe.ru
dev.en	dev.ru
dev.tok.en	dev.tok.ru
test.en	test.ru

Пример корпуса, состоящего из tmx-файла и нескольких текстовых файлов для англо-русского направления:

testcorpus.tmx
testcorpus1.en
testcorpus1.ru
testcorpus2.en
testcorpus2.ru

6. Обучение модели

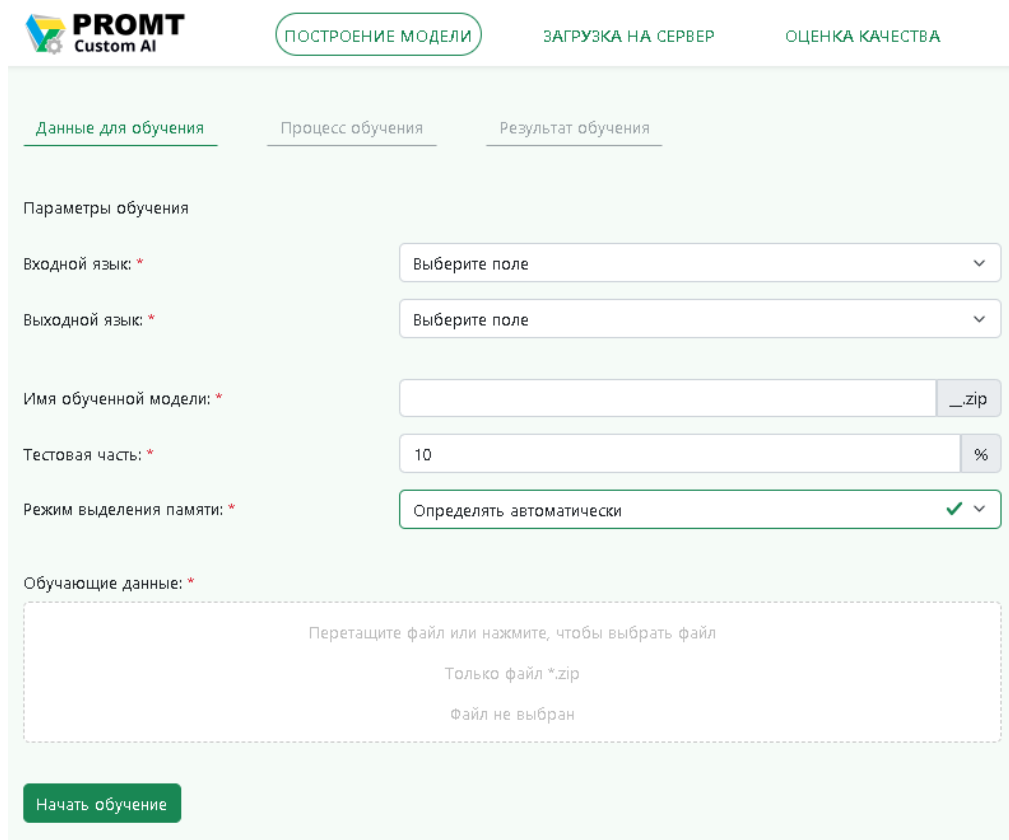
В качестве исходных данных для обучения модели используется базовая модель* и обучающие данные в виде архива формата zip.

* NMT-модели устанавливаются в составе языковых пакетов на этапе установки PROMT Custom AI или позднее, с помощью скрипта `install_models.sh` (см. разделы [Установка продукта](#) и [Установка языковых пакетов](#)).

 Подробнее об обучающих данных см. в разделе [Подготовка исходных данных](#).

6.1. Задание параметров обучения и запуск процесса обучения

Запуск процесса обучения модели осуществляется на вкладке *Данные для обучения* страницы *Построение модели*.



The screenshot shows the 'Построение модели' (Building Model) page of the PROMT Custom AI interface. The 'Данные для обучения' (Data for Training) tab is active. The page contains several input fields for training parameters:

- Входной язык:** * (Input language) - dropdown menu with 'Выберите поле' (Select field).
- Выходной язык:** * (Output language) - dropdown menu with 'Выберите поле' (Select field).
- Имя обученной модели:** * (Trained model name) - text input field with a '_zip' suffix.
- Тестовая часть:** * (Test part) - text input field with '10' and a '%' suffix.
- Режим выделения памяти:** * (Memory allocation mode) - dropdown menu with 'Определять автоматически' (Determine automatically) and a green checkmark.
- Обучающие данные:** * (Training data) - a dashed box containing the text 'Перетащите файл или нажмите, чтобы выбрать файл' (Drag and drop file or click to select file), 'Только файл *.zip' (Only *.zip file), and 'Файл не выбран' (File not selected).

At the bottom left, there is a green button labeled 'Начать обучение' (Start training).

Рис. 6.1 Страница *Данные для обучения*

 Чтобы начать обучение модели, выполните следующие шаги:

1. Задайте параметры обучения в соответствующих полях:

- *Входной язык*

Выберите необходимый входной язык из списка доступных языков.

 Список доступных языков ограничивается файлом лицензии.

- *Выходной язык*

Выберите необходимый выходной язык из списка доступных языков.

 Список доступных языков ограничивается файлом лицензии.

- *Имя обученной модели*

Задайте название для настроенной модели.

 Имя модели должно быть уникальным и может содержать только буквы латинского алфавита, цифры, дефисы и нижнее подчеркивание.

 По умолчанию в название настроенной модели автоматически добавляются префиксы указанных вами входного и выходного языков. Отключить данную опцию можно в параметре *AddPrefixToModelName* конфигурационного файла PAI (по умолчанию `/usr/local/pai1.1/addon/app/appsettings.json`).

- *Тестовая часть*

Укажите суммарный объем корпуса для валидации и тестового корпуса в процентах от исходного корпуса.

 Рекомендуемое значение для данного параметра – от 1 до 10%. Суммарное максимальное число строк, выделяемое для этих двух корпусов, не может превышать 6000. При этом исходный обучающий корпус уменьшается на соответствующее количество строк. По умолчанию тестовая часть составляет 10%.

- *Режим выделения памяти*

По умолчанию размер памяти, выделяемой для обучения, определяется автоматически. Выделяемый объем вычисляется перед запуском настройки модели, исходя из доступной памяти GPU-устройства/ устройств и из размера базовой модели.

Чтобы задать свое значение, выберите опцию *Задать свое значение* и укажите объем памяти в мегабайтах:

Параметры обучения

Входной язык: * Английский ✓

Выходной язык: * Русский ✓

Имя обученной модели: * trained _en_ru.zip

Тестовая часть: * 10 %

Режим выделения памяти: * ➞ Задать свое значение ✓

Выделение памяти: * ➞ 0 Мб

💡 Рекомендуется выделять максимальное количество памяти, оставляя при этом свободными ~ 2.5 Гб на GPU-устройстве.

⚠ Если выделенный объем памяти слишком большой, процесс обучения будет остановлен с сообщением об ошибке.

- **Базовая модель**

Опция выбора базовой модели доступна только в том случае, если на выбранном языковом направлении есть более одной модели. Выберите модель из списка доступных моделей указанного языкового направления в выпадающем списке:

Параметры обучения

Входной язык: * Русский ✓

Выходной язык: * Английский ✓

Имя обученной модели: * trained _ru_en.zip

Тестовая часть: * 10 %

Режим выделения памяти: * Определять автоматически ✓

Базовая модель: * ➞ Выберите поле

Проводить скоринг обучающих данных: basic

Обучающие данные: * new

✿ В выпадающем списке отображаются только те языковые модели, которые соответствуют текущему выбранному входному и выходному языку. В списке доступны модели из языковых пакетов, установленных в процессе установки продукта или при помощи скрипта `install_models.sh` (см. [Установка языковых пакетов](#)).

Для англо-русского и русско-английского направления при наличии модели на обратном языковом направлении становится доступной опция *Проводить скоринг обучающих данных*:

Режим выделения памяти: * Определять автоматически ✓

Базовая модель: * basic ✓

Проводить скоринг обучающих данных: ☒ ←

По умолчанию опция отключена, и скоринг при обучении модели не выполняется.

Включите данную опцию, если необходимо, чтобы при обучении модели выполнялся скоринг обучающих данных с использованием обратного языкового пакета.

! Скоринг обучающих данных доступен только на англо-русском и русско-английском направлении.

Если на обратном языковом направлении установлено более одной модели, при включении опции *Проводить скоринг обучающих данных* появляется поле *Обратная модель* с выпадающим списком моделей. Выберите из списка модель, которая будет использоваться для скоринга:

Проводить скоринг обучающих данных: ☒

Обратная модель: Выберите поле

Обучающие данные: *

Перетащите файл или нажмите, чтобы выбрать файл

Только файл *.zip

В списке обратных моделей отображаются только те языковые модели, которые соответствуют текущему выбранному входному и выходному языку.

***** Если включена опция *Проводить скоринг обучающих данных*, но модель не задана явно, для скоринга будет использоваться первая модель из списка моделей, полученных от сервера.

- **Обучающие данные**

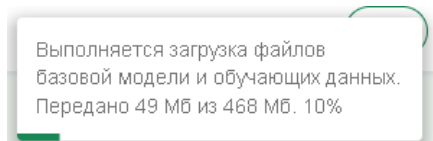
Чтобы загрузить архив с обучающими данными, щелкните мышью в поле *Обучающие данные* и выберите необходимый файл или перетащите файл архива с обучающими данными в это поле.

***** В поле *Обучающие данные* может быть загружен только файл формата .zip.

2. Нажмите кнопку *Начать обучение*, чтобы запустить процесс обучения модели.

***** Если все настройки указаны верно, будет произведена загрузка данных на сервер и начнется процесс обучения модели.

Статус загрузки данных при старте обучения выводится в правом верхнем углу веб-интерфейса:



✳ В том случае если в лицензии указан неверный hwid, обучение остановится с ошибкой. При этом если в PAI настроены почтовые уведомления, пользователи, указанные в качестве получателей, получат письмо *Обучение завершено с ошибкой*.

6.2. Процесс обучения

После старта процесса обучения откроется вкладка *Процесс обучения*, на которой выводится лог процесса обучения модели:

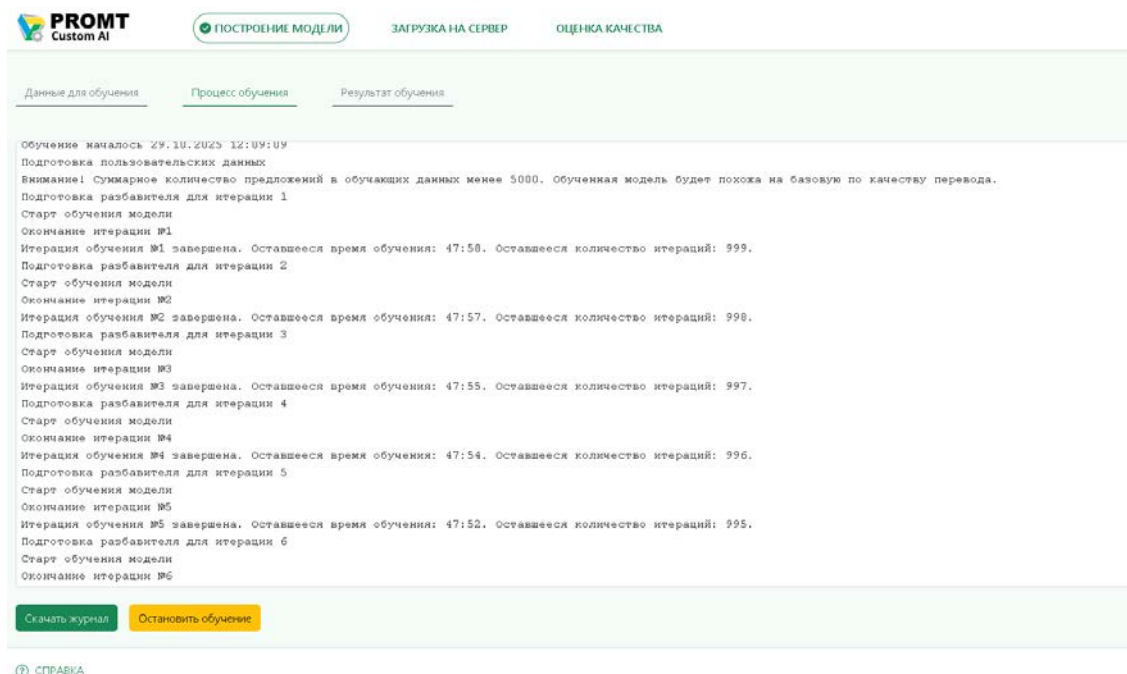
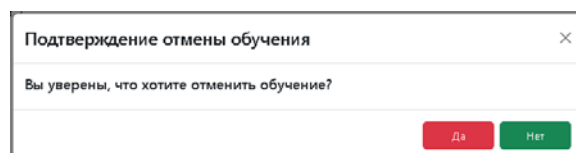


Рис. 6.5 Вкладка *Процесс обучения*

- Чтобы скачать лог с информацией о текущих результатах обучения, нажмите на кнопку *Скачать журнал*.

✿ Лог текущего обучения хранится в файле `last_training_log.log` в папке `logs` (по умолчанию `/usr/local/pai1.1/addon/logs`).

- Чтобы прервать обучение модели, нажмите на кнопку *Остановить обучение*. Подтвердите действие в открывшемся диалоге *Подтверждение отмены обучения*:



✿ Если на момент остановки пользователем обучения была завершена хотя бы одна итерация обучения модели, откроется вкладка *Результат обучения*, на которой будет доступна для скачивания обученная модель.

✳ В случае если в цикле обучения модели произошла ошибка, обучение будет остановлено с сообщением об ошибке. При этом если на момент остановки обучения была завершена хотя бы одна итерация обучения, откроется вкладка *Результат обучения*, на которой будет доступна для скачивания обученная модель.

✳ Если в PAI включены почтовые уведомления, пользователям, которые указаны в файле конфигурации в качестве получателей, будет отправлено соответствующее уведомление в случае успешного окончания обучения, в случае остановки обучения пользователем, а также в том случае, если до начала обучения произошла ошибка. Настроить уведомления можно в конфигурационном файле PAI (см. [Настройка почтовых уведомлений](#)).

6.2.1. Этапы обучения модели

Процесс обучения модели может быть представлен в виде двух этапов: препроцессинг текстового корпуса и непосредственно обучение модели.

Препроцессинг состоит из следующих шагов:

1. Создание исходного параллельного текстового корпуса по папке с входными файлами
Выполняется утилитой `prompt.nts.extractcorpus.exe`. На вход утилите передается путь к папке с текстовым корпусом. На выходе получается пара файлов `corpus.$src` и `corpus.$tgt`, в которых объединены все текстовые сегменты для входа и выхода из исходной папки.
2. Нормализация пунктуации и токенизация пользовательского корпуса
Производится утилитой `prompt.nts.tokenize`. На вход подается корпус `corpus`, на выходе получается токенизированный корпус `corpus.tok`.
3. Прочистка пользовательского корпуса
Производится утилитой `prompt.nts.cleancorpus`.
 - a. Основная прочистка
Из сета `corpus` исключаются пустые, дублирующиеся или слишком длинные строки. Максимальная длина строки на входе и выходе не должна превышать 100 слов и 1000 символов. В процессе работы утилита `prompt.nts.cleancorpus.exe` отфильтровывает и сохраняет в отдельный лог `corpus.tok.def.clean.badstring.txt` строки, которые не могут быть использованы для обучения модели. В логе указывается причина фильтрации, входная строка и выходная строка.

Например:

```
Bad length:
Sberbank 's digital solutions are among the most widely used in the world .
компаний
```

Ниже приведены причины, по которым строки могут попасть в лог:

Сообщение в логе	Причина попадания в лог
------------------	-------------------------

Duplicate	Пара вход-выход уже встречалась ранее
Equal lines	Вход = выход
Bad length	Вход или выход имеет меньше 1 слова (пустой), больше 100 слов, либо менее 10 символов
Bad alphanumeric % in source/target	Во входной/ выходной строке кол-во букв < 30% от всего входа
Source/Target is empty	После выполнения обработки во входной/выходной строке не осталось символов
Source or target is too long or too short	После выполнения обработки во входной/выходной строке стало меньше 1 или больше 100 слов
Bad source length/target length ratio	После выполнения обработки отношение длин входного текста к выходному > 9 или наоборот, если отношение выходного к входному > 9.

b. Языковая прочистка

Из сета corpus удаляются пары строк, которые написаны на других языках

c. Скоринг нейронными моделями*

* Скоринг выполняется в том случае, если перед стартом обучения была загружена обратная модель (опция доступна для англо-русского и русско-английского направлений).

Для запуска скоринга пользовательских данных нейронными моделями используется marian.exe. Запуск marian.exe выполняется утилитой prompt.nts.MarianLauncher.exe.

4. Подготовка разбавителя

После препроцессинга пользовательского корпуса выполняется подготовка разбавителя. Из корпуса общелексических данных (разбавителя) извлекается случайная выборка, равная по объему пользовательскому корпусу. Затем выборка токенизируется, строится выравнивание, и выборка из пользовательского корпуса объединяется с выборкой из разбавителя. Подготовка разбавителя происходит в начале каждой итерации обучения.



Корпус общелексических данных входит в языковой пакет в составе набора.

5. Создание корпуса для валидации данных и тестового корпуса

Выполняется утилитой prompt.nts.extractrandomset. По заданному проценту извлекаемых сегментов (Validation_Percent) утилита выбирает случайные сегменты из исходного корпуса. Максимальное число строк в случайной выборке ограничено значением 6000. Полученная случайная выборка строк равномерно распределяется между создаваемыми новыми корпусами: корпусом для валидации (dev-сет) и тестовым корпусом (test-сет). Результат записывается в файлы dev.tok.\$src, dev.tok.\$tgt и test.tok.\$src, test.tok.\$tgt.

Файлы `dev.tok.$src` и `dev.tok.$tgt` используются для валидации данных в процессе настройки модели.

Для тестового токенизированного корпуса `test.tok` выполняется операция детокенизации, результат записывается в файлы `test.$src` и `test.$tgt`.

Файлы `test.$src` и `test.$tgt` используются для оценки BLEU после завершения настройки модели.

Из исходного корпуса удаляются строки, попавшие в случайную выборку, т.о. обучение происходит на исходном корпусе минус корпус для валидации и тестовый корпус.

6. Генерация синтетических данных с плейсхолдерами для незнакомых слов для пользовательского корпуса

Производится утилитой `prompt.nts.makeunkdata`. По сету `corpus` строится дополнительный сет, содержащий пометки `<unk>` вместо случайно взятых слов. Дополнительный сет записывается в файлы с расширением `.sub`.

7. Токенизация тестовых, валидационных и синтетических данных

Производится утилитой `prompt.nts.tokenize`. На выходе получаются обработанные файлы основного сета, синтетического сета и сета `dev` с дополнительной строкой `.bpe` в названии.

8. Построение выравнивания для пользовательских данных

Выполняется утилитой `prompt.nts.makeunkdata`. На вход подается объединенный корпус, результат работы записывается в файл `aligned-grow-diag-final.realigned`.

После подготовки и обработки исходных данных запускается процесс **обучения** NMT-модели с помощью приложения командной строки `marian.exe`. Запуск `marian.exe` выполняется утилитой `prompt.nts.MarianLauncher.exe`.

Утилита `prompt.nts.MarianLauncher.exe`:

1. Распаковывает архив с базовой моделью во временную папку. Временная папка создается в той же папке, где находится базовая модель.

✳ После завершения обучения временная папка удаляется.

На этом же этапе выполняется распаковка архива с обратной моделью (если в настройках обучения была указана обратная модель).

2. Подготавливает распакованные файлы архива к последующей настройке модели:

- a. Находит файл модели `"model.npz.best-bleu.npz"` и переименовывает в `"model.baseline.npz"`;

- b. Находит файл настроек модели `"model.npz.best-bleu.npz.decoder.yml"`, устанавливает в нем значение `"alignement: 0"`, переименовывает в `"model.baseline.npz.decoder.yml"`.

3. Вычисляет значения параметра `dim-vocab`, анализируя файлы `vocab.src.yml` и `vocab.tgt.yml`. Извлекает из `model.baseline.npz` файл `special.model.yml`, из которого считывается информация об архитектуре* модели: `transformer-base` или `transformer-big`.

* В зависимости от архитектуры при обучении используются разные наборы параметров.

4. При необходимости автоматически вычисляет значения параметров *"Model_Workspace"* и *"Model_Valid_Freq"*.
5. После успешного завершения обучения модели формирует файл архива с результирующей моделью и записывает его по пути, указанном в параметре *"ResultModelsDirectory"* файла конфигурации *appsettings.json* (по умолчанию */usr/local/pai1.1/addon/app/appsettings.json*).



По умолчанию в качестве результирующей модели сохраняется лучшая модель.



После завершения обучения происходит конвертация модели в ct2, если соответствующий параметр указан в файле конфигурации *appsettings.json* (по умолчанию конвертация включена). Конвертация выполняется утилитой *prompt.nts.ct2converter.exe*.



Для конвертации настраиваемых моделей marian в CT2 необходимым условием является наличие Python и установленных для него пакетов.

6.3. Результат обучения

По завершении процесса обучения модели откроется вкладка *Результат обучения*.

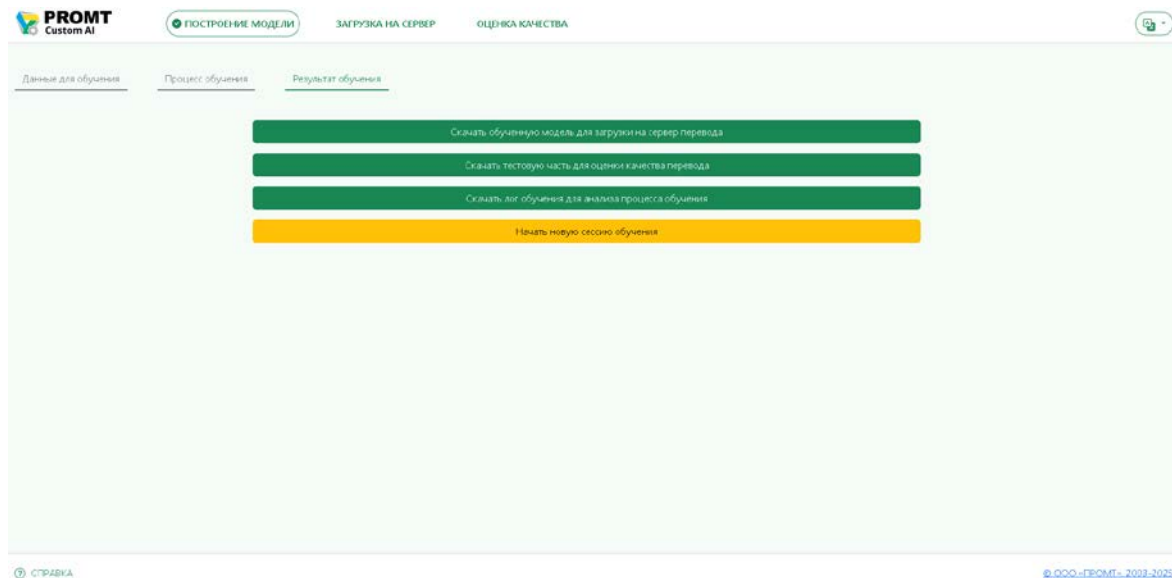


Рис. 6.6 Вкладка *Результат обучения*

- Чтобы скачать обученную модель, выберите кнопку *Скачать обученную модель для загрузки на сервер перевода*

Модель будет скачана в виде файла zip. Скачанная из *PAI* модель может быть затем загружена в *PNTS* или *PTF* (см. [Загрузка настроенной модели на сервер](#)).

- Чтобы скачать тестовую часть, выберите кнопку *Скачать тестовую часть для оценки качества перевода*

Тестовая часть будет скачана в виде архива `test_data.zip`. Архив содержит файлы `test.$src` и `test.$trg`, которые могут быть загружены на странице *Оценка качества* для получения метрики BLEU (см. раздел [Оценка результатов](#)).

- Чтобы скачать лог обучения, выберите кнопку *Скачать лог обучения для анализа процесса обучения*.

Лог будет скачан в виде архива `log_data.zip`.

Архив содержит следующие файлы:

`corpus.tok.def.clean.badstring.txt` – отфильтрованные строки, которые не были использованы для обучения модели;

`<modelName>.zip.log` – полный лог обучения;

`<modelName>_marian.train.log` – лог обучения marian;

`<modelName>_marian.valid.log` – лог валидации.

- Чтобы запустить новое обучение, выберите кнопку *Начать новую сессию обучения*. Подтвердите действие в открывшемся диалоге *Подтверждение начала новой сессии обучения*.

 Перед началом новой сессии следует сохранить все необходимые данные, воспользовавшись кнопками на странице *Результат обучения*. В случае если результирующая модель не была сохранена по кнопке *Скачать обученную модель для загрузки на сервер перевода*, по умолчанию вы можете найти её в папке продукта (по умолчанию `/usr/local/pai1.1/addon/storage/training/results`). Путь к папке с обученной моделью указывается в параметре `"ResultModelsDirectory"` конфигурационного файла `appsettings.json` (по умолчанию `/usr/local/pai1.1/addon/app/appsettings.json`).

 Если в PAI включены почтовые уведомления, по завершении обучения пользователи, указанные в качестве получателей, получают соответствующее уведомление. Подробнее о почтовых уведомлениях в PAI см. в разделе [Настройка почтовых уведомлений](#).

7. Загрузка настроенной модели на сервер

Обученная модель может быть выгружена из PAI и затем загружена в PNTS или PTF.

Загрузка настроенной модели в PNTS или PTF осуществляется на странице *Загрузка на сервер*.

Рис. 7.1 Страница загрузки модели на сервер

 Чтобы загрузить настроенную модель на сервер PNTS или PTF, выполните следующую последовательность действий:

1. В поле *Тип загружаемой модели* выберите одно из следующих значений:

- **Выбрать модель** (установлено по умолчанию)

Чтобы выбрать модель, щелкните мышью в поле *Файл модели* и выберите необходимый файл с архивом модели или перетащите файл в данное поле.

Затем укажите входной и выходной языки в поле *Входной язык* и в поле *Выходной язык*.

- **Последняя обученная модель**

✿ Опция доступна только в том случае, если в текущей сессии было успешно завершено обучение модели.

При выборе данной опции указывать входной и выходной язык не требуется – на сервер будет загружена последняя обученная модель.

2. Укажите название профиля, в который будет загружена настроенная модель

✿ Название профиля должно быть уникальным. При загрузке будет создан новый профиль с указанным именем, включающий в себя загруженную модель. Более подробную информацию о профилях перевода вы найдете в Руководстве администратора PNTS.

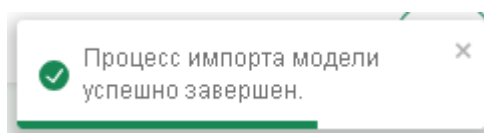
3. Нажмите кнопку *Загрузить модель на сервер*

✿ Поля *Имя пользователя* и *Пароль* опциональны, их необходимо заполнять только при подключении к PNTS Windows-версии.

❗ В поле *Адрес сервера* прописан адрес, указанный при установке PAI. Заменить адрес сервера можно только в файле конфигурации `appsettings.json` (по умолчанию `/usr/local/pai1.1/addon/app/appsettings.json`) в секциях "*BleuUtility*" и "*PntsProxy*". После уточнения файла конфигурации необходимо перезапустить сервис `pai` (см. раздел [Активация](#)).

🟡 Начнется процесс загрузки модели на сервер.

После успешной загрузки модели на сервер в правой верхней части окна приложения будет показано следующее сообщение:



Если после начала загрузки выводится сообщение о сбое при загрузке модели на сервер, проверьте логи PAI (по умолчанию логи находятся в папке `/usr/local/pai1.1/addon/logs`) и логи PNTS / PTF (см. Руководство администратора PNTS и Руководство администратора PTF, соответственно).

8. Оценка результатов

Качество обученной модели оценивается с помощью вычисленного значения метрики BLEU, которое можно получить на странице *Оценка качества* после того, как настроенная модель была загружена на сервер.

PROMT
Custom AI

ПОСТРОЕНИЕ МОДЕЛИ ЗАГРУЗКА НА СЕРВЕР **ОЦЕНКА КАЧЕСТВА**

Входной язык: * Английский ✓ ▾

Выходной язык: * Русский ✓ ▾

Профиль (базовая модель): * General Domain ✓ ▾

Профиль (обученная модель): * new ✓ ▾

Адрес сервера: * https://172.17.6.8/pnts

Имя пользователя:

Пароль:

Файл для перевода: *

Перетащите файл или нажмите, чтобы выбрать файл

Укажите тестовую часть на входном языке

test.en

Файл эталонного перевода: *

Перетащите файл или нажмите, чтобы выбрать файл

Укажите тестовую часть на выходном языке

test.ru

Посчитать метрику BLEU

Рис. 8.1 Страница *Оценка качества*

▶▶ Чтобы оценить результаты настройки модели, выполните следующую последовательность действий:

1. Выберите входной язык в поле *Входной язык*
2. Выберите выходной язык в поле *Выходной язык*
3. Выберите профиль с базовой моделью в поле *Профиль (базовая модель)*
4. Выберите профиль с обученной моделью в поле *Профиль (обученная модель)*

❗ В поле *Адрес сервера* прописан адрес, указанный при установке PAI. Заменить адрес сервера можно только в файле конфигурации `appsettings.json` (по умолчанию `/usr/local/pai1.1/addon/app/appsettings.json`) в секциях "*BleuUtility*" и "*PntsProxy*". После уточнения файла конфигурации необходимо перезапустить сервис `pai` (см. [Активация](#)).

✳ Поля *Имя пользователя* и *Пароль* опциональны, их следует заполнять только при подключении к PNTS Windows-версии.

5. Укажите файл для перевода

Щелкните мышью в поле *Файл для перевода* и выберите необходимый файл или перетащите необходимый файл в данное поле.

✳ В качестве файла для перевода могут быть загружены файлы `.$src` и `.txt` (utf-8).

💡 Если вы хотите использовать для оценки результатов файлы из тестовой части, скачанной на странице *Результат обучения*, необходимо предварительно извлечь файлы `test.$src` и `test.$trg` из скачанного архива.

6. Укажите файл эталонного перевода

Щелкните мышью в поле *Файл эталонного перевода* и выберите необходимый файл или перетащите необходимый файл в данное поле.

✳ В качестве файла эталонного перевода могут быть загружены файлы `.$trg` и `.txt` (utf-8).

💡 Если вы хотите использовать для оценки результатов файлы из тестовой части, скачанной на странице *Результат обучения*, необходимо предварительно извлечь файлы `test.$src` и `test.$trg` из скачанного архива.

✳ Указанные параметры запуска подсчета BLEU сохраняются на странице *Оценка качества* в рамках текущей сессии.

7. Нажмите кнопку *Посчитать метрику BLEU*

🟡 Начнется процесс вычисления BLEU.

Сначала будет выполнена обработка файла для перевода с базовым профилем и вычислено значение BLEU для базовой модели, затем будет выполнена обработка файла для перевода с профилем с настроенной моделью и вычислено значение BLEU для настроенной модели.

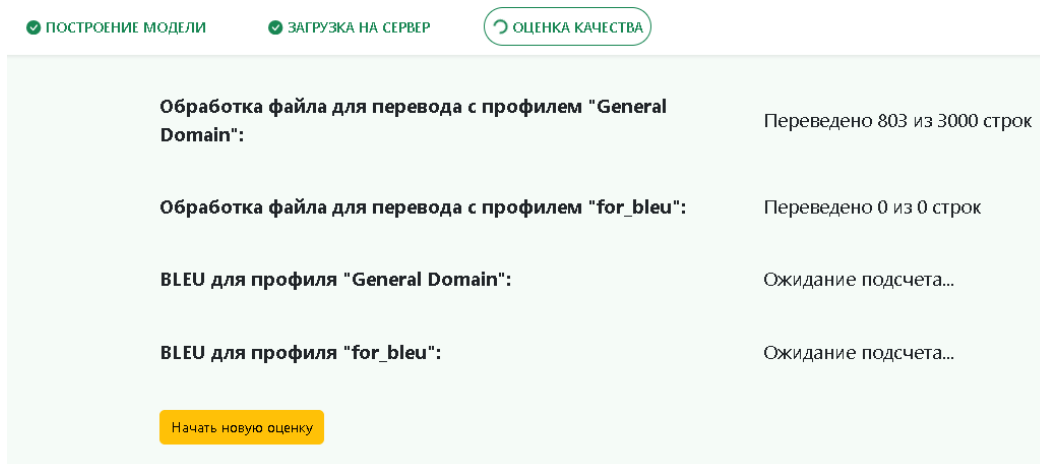


Рис. 8.2 Процесс вычисления BLEU

По окончании обработки переводы, выполненные профилем с базовой моделью и профилем с обученной моделью, могут быть скачаны в виде текстовых файлов по кнопке *Скачать перевод*.

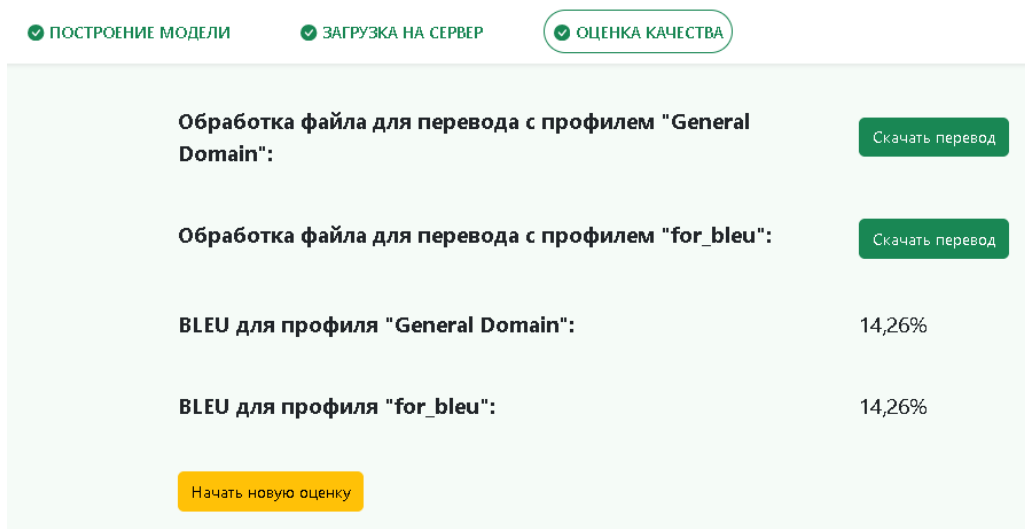


Рис. 8.3 Результаты BLEU

Если значение BLEU, полученное при переводе с обученной моделью, больше, чем значение BLEU, полученное при переводе с базовой моделью, то результат настройки модели считается успешным.

Приложение 1. Глоссарий

BLEU (Bilingual Evaluation Understudy) – измерение различий между автоматическим переводом и одним или несколькими эталонными пользовательскими переводами одного исходного предложения. Алгоритм BLEU сравнивает последовательные фразы автоматического перевода с последовательными фразами, которые он находит в эталонном переводе, и взвешенно подсчитывает количество совпадений. Эти совпадения не зависят от позиции. Высшая степень совпадения указывает на более высокую степень сходства с эталонным переводом и более высокий балл. Внятность и грамматика не учитываются.

Апдейт (update) – обновление весов при обучении.

Валидация – перевод dev-сета с вычислением BLEU.

Итерация – полный проход по всему корпусу обучающих данных за один цикл обучения модели. Один проход по всему циклу работы PAI включает в себя выделение новой части данных из разбавителя, её обработку и формирование нового обучающего корпуса для запуска новой итерации обучения.

Корпус для валидации (dev set). Из входного корпуса после токенизации и очистки случайным образом выбираются строки (заданный процент от общего числа, максимальное количество – 6000 строк). Выбранные строки удаляются из входного корпуса. Половина из выбранных строк записывается в новый корпус, называемый **корпусом для валидации** (dev.tok.\$src, dev.tok.\$tgt). Вторая половина записывается в *тестовый корпус*. Модель обучается на строках "входной корпус минус корпус для валидации и минус тестовый корпус". После каждого цикла обучения делается перевод корпуса для валидации и считаются показатели эффективности обучения (BLEU и др.), которые оцениваются после каждого цикла, и, если их прирост значительно замедляется или останавливается, обучение будет окончено.

Тестовый корпус для подсчета финального BLEU (test set) . Из входного корпуса после токенизации и очистки случайным образом выбираются строки (заданный процент от общего числа, максимальное количество – 6000 строк). Выбранные строки удаляются из входного корпуса. Половина из выбранных строк записывается в *корпус для валидации*. Вторая половина записывается в новый корпус, называемый **тестовым корпусом** (test.\$src, test.\$tgt). После завершения обучения выполняется перевод тестового корпуса базовой моделью и настроенной моделью и вычисляются BLEU для каждой модели.

Минибатч – количество предложений, которое просматривает модель при обучении за один апдейт.

Модель нейронного машинного перевода (NMT-модель) – лингвистические данные, необходимые для выполнения нейронного машинного перевода (NMT).

Нормализация пунктуации – приведение знаков пунктуации к последовательности &<code>, экранирование.

Построение выравнивания – соотнесение токенов из входной строки с токенами из выходной строки и запись результата в формате, понятном для `mapan` (например, 1-1 2-4 и т.д.).

Профиль перевода – совокупность сохраненных лингвистических настроек, которые доступны во всех приложениях переводчика PROMT и позволяют настраивать его на перевод текстов определенной предметной области. В некоторых продуктах PROMT называется также тематикой.

Разбавитель – корпус общелексических данных, который входит в языковой пакет в составе набора. В начале каждой итерации обучения из разбавителя извлекается случайная выборка, равная по объему пользовательскому корпусу. Затем выборка токенизируется, строится выравнивание, и выборка из пользовательского корпуса объединяется с выборкой из разбавителя.

Сегментация BPE и обработка регистра. Входные данные приводятся в формат, с которым работает `mapan`. **Сегментация BPE** – процесс разбиения слов на сегменты по словарю BPE, **обработка регистра** – приведение к нижнему регистру с сохранением информации о начальном состоянии (например, |С - начиналось с верхнего регистра, |U - было полностью в верхнем регистре).

Скоринг – процесс оценки сегментов параллельного корпуса на соответствие друг другу нейронными моделями. Чем оценка выше, тем лучше сегменты соотносятся друг с другом по содержанию и форме. Низкая оценка говорит о том, что в паре сегментов есть расхождения (например, лишний текст в одном из сегментов или полное расхождение по содержанию). Пары сегментов, которые по результатам скоринга получили оценку ниже заданного порога, удаляются из корпуса.

Токен – слово или знак пунктуации, обрамленный пробелами.

Токенизация – разделение строки на токены, которыми являются слова, знаки пунктуации и неалфавитные символы. Все токены отделяются друг от друга символом пробела.

Языковая пара – пара «входной язык – выходной язык». Языковая пара указывает, с какого языка и на какой будет переводиться текст.

Приложение 2. Префиксы для обозначения языков

В PNTS, PTF и PAI для обозначения языков используются префиксы в формате RFC. В PAI языковые префиксы используются для обозначения входного и выходного языка в названиях файлов текстового корпуса и в названиях моделей. Помимо этого, префиксы используются для идентификации языка при запуске обучения моделей и при запуске утилиты для подсчета BLEU.

❗ Для моделей с китайским языком при подсчете BLEU следует использовать префиксы zh-tw и zh-cn, а для запуска обучения моделей – префикс zh.

❗ Для моделей с кириллическим сербским префикс sr-cyrl используется только в названиях моделей.

язык	префикс
Азербайджанский	az
Английский	en
Арабский	ar
Армянский	hy
Белорусский	be
Бирманский	my
Болгарский	bg
Венгерский	hu
Вьетнамский	vi
Греческий	el
Грузинский	ka
Дари	fa-af
Датский	da
Иврит	he
Индонезийский	id
Испанский	es
Итальянский	it
Казахский	kk
Киргизский	ky
Китайский (традиционный)	zh-tw
Китайский (упрощенный)	zh-cn
Корейский	ko
Курдский	ku
Лаосский	lo

Латышский	lv
Литовский	lt
Малайский	ms
Монгольский	mn
Немецкий	de
Нидерландский	nl
Норвежский	nb
Нюнорск	nn
Польский	pl
Португальский	pt
Пушту	ps
Румынский	ro
Русский	ru
Сербский	sr
Сербский (Кириллица)	sr-cyrl
Словацкий	sk
Суахили	sw
Тагальский	tl
Таджикский	tg
Тайский	th
Татарский	tt
Турецкий	tr
Туркменский	tk
Узбекский	uz
Украинский	uk
Урду	ur
Фарси	fa
Финский	fi
Французский	fr
Хинди	hi
Хорватский	hr
Чешский	cs
Чувашский	cv
Шведский	sv
Эстонский	et
Японский	ja

Приложение 3. Установка необходимых компонентов для Python на системах Linux

Пререквизиты, необходимые для работы PAI, включены в состав дистрибутива.

Распакуйте архив инсталляционного набора, переданный вам при покупке программы.

Например

```
sudo unzip ./PAI1.1_PR24_0115.zip
```

✳️ Результатом выполнения команды будет установочный пакет, распакованный в текущую папку.

Папка с распакованным архивом содержит установочный run-файл и папку с пререквизитами.

Офлайн-установка необходимых компонентов выполняется из папки с пререквизитами:
<папка с распакованным установочным пакетом>/Prerequisites/python3.

⚠️ Установите все доступные для вашей ОС обновления перед началом установки.

⚠️ На системах, использующих venv, для установки venv требуется подключение к Интернету.

Ubuntu 20.04

Python включен в дистрибутив Ubuntu 20.04, поэтому установка самого Python на данных ОС не требуется. Необходимо установить pip и пакеты для Python.

Установка при наличии подключения к интернету:

1. Обновите список пакетов

```
sudo apt update
```

2. Установите pip

```
sudo apt install python3-pip
```

3. Установите пакеты

```
sudo python3 -m pip install ctranslate2==2.23.0
```

✳️ При установке пакета ctranslate2 будут автоматически установлены подходящие версии пакетов PyYAML и numpy.

Установка в режиме офлайн:

1. Установите distutils

```
sudo dpkg -i /path/to/prerequisitesFolder/python3/Ubuntu20.04/python3-distutils_3.8.10-0ubuntu1~20.04_all.deb
```

✳️ /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы

2. Установите pip:

```
sudo python3 /path/to/prerequisitesFolder/python3/pip-23.0.1-py3-none-any.whl/pip
install /path/to/Prerequisites/python3/pip-23.0.1-py3-none-any.whl
```

✳️ /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы

3. Установите пакеты:

numpy

```
sudo python3 -m pip install
/path/to/prerequisitesFolder/python3/Ubuntu20.04/numpy-1.24.2-cp38-cp38-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

ctranslate2

```
sudo python3 -m pip install
/path/to/prerequisitesFolder/python3/Ubuntu20.04/ctranslate2-2.14.0-cp38-cp38-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

✳️ /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

Ubuntu 22.04

Python включен в дистрибутив Ubuntu 22.04, поэтому установка самого Python на данных ОС не требуется. Необходимо установить pip и пакеты для Python.

Установка при наличии подключения к интернету:

1. Обновите список пакетов

```
sudo apt update
```

2. Установите pip

```
sudo apt install python3-pip
```

3. Установите пакеты

```
sudo python3 -m pip install ctranslate2==2.23.0
```

✳️ При установке пакета ctranslate2 будут автоматически установлены подходящие версии пакетов PyYAML и numpy.

Установка в режиме офлайн:

1. Установите pip

```
sudo python3 /path/to/prerequisitesFolder/python3/pip-23.0.1-py3-none-any.whl/pip
install /path/to/Prerequisites/python3/pip-23.0.1-py3-none-any.whl
```

✳️ /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

2. Установите пакеты

numpy

```
sudo python3 -m pip install
/path/to/prerequisitesFolder/python3/Ubuntu22.04/numpy-1.24.2-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

PyYAML

```
sudo python3 -m pip install
/path/to/prerequisitesFolder/python3/Ubuntu22.04/PyYAML-6.0-cp310-cp310-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.
whl
```

ctranslate2

```
sudo python3 -m pip install
/path/to/prerequisitesFolder/python3/Ubuntu22.04/ctranslate2-2.14.0-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

✳️ /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

Ubuntu 24.04

Python включен в дистрибутив Ubuntu 24.04, поэтому установка самого Python на данных ОС не требуется. Необходимо установить venv и пакеты для Python.

✳️ При установке набора на Ubuntu 24.04 необходимо указывать python3 из виртуального окружения /path/to/venv/bin/python3, в противном случае установленные пакеты находиться не будут.

Для установки необходимых компонентов выполните следующую последовательность действий:

1. Обновите список пакетов

```
sudo apt update
```

2. Установите venv

```
sudo apt install python3-venv
```

3. Создайте папку для venv

```
sudo mkdir /path/to/venv
```

✳️ /path/to/venv – путь к папке, в которой будет развернут venv

4. Создайте venv

```
sudo python3 -m venv /path/to/venv
```

Например,

```
sudo python3 -m venv /usr/local/venv
```

5. Выдайте права на папку для venv

```
sudo chmod -R a+rwX /path/to/venv
```

6. Активируйте venv

```
source /path/to/venv/bin/activate
```

7. Установите пакеты

numpy

```
python3 -m pip install /path/to/prerequisitesFolder/python3/Ubuntu24.04/numpy-2.1.3-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

PyYAML

```
python3 -m pip install /path/to/prerequisitesFolder/python3/Ubuntu24.04/PyYAML-6.0.2-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

ctranslate2

```
python3 -m pip install /path/to/prerequisitesFolder/python3/Ubuntu24.04/ctranslate2-2.23.0-cp312-cp312-linux_x86_64.whl
```

✳️ /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

8. Выйдите из venv

```
deactivate
```

Debian 11

Python включен в дистрибутив Debian 11, поэтому установка самого Python на данной ОС не требуется. Необходимо установить pip и пакеты для Python.

Установка при наличии подключения к интернету:

1. Обновите список пакетов

```
sudo apt update
```

2. Установите pip

```
sudo apt install python3-pip
```

3. Установите пакеты

```
sudo python3 -m pip install ctranslate2==2.23.0
```

Установка в режиме офлайн:

1. Установите distutils

```
sudo dpkg -i /path/to/prerequisitesFolder/python3/Debian\ 11/python3-distutils_3.9.2-1_all.deb
```

 /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

2. Установите pip:

```
sudo python3 /path/to/prerequisitesFolder/python3/pip-23.0.1-py3-none-any.whl/pip install /path/to/Prerequisites/python3/pip-23.0.1-py3-none-any.whl
```

 /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

3. Установите пакеты:

numpy

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/Debian\ 11/numpy-1.21.6-cp39-cp39-manylinux_2_12_x86_64.manylinux2010_x86_64.whl
```

PyYAML

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/Debian\ 11/PyYAML-6.0-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2010_x86_64.whl
```

ctranslate2

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/Debian\ 11/ctranslate2-2.14.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

 /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

Debian 12

Python включен в дистрибутив Debian12, поэтому установка самого Python на данной ОС не требуется. Необходимо установить venv и пакеты для Python.

 При установке набора на Debian12 необходимо указывать python3 из виртуального окружения /path/to/venv/bin/python3, в противном случае установленные пакеты находиться не будут.

Для установки необходимых компонентов выполните следующую последовательность действий:

1. Обновите список пакетов

```
sudo apt update
```

2. Установите venv

```
sudo apt install python3-venv
```

3. Создайте папку для venv

```
sudo mkdir /path/to/venv
```



/path/to/venv – путь к папке, в которой будет развернут venv

4. Создайте venv

```
sudo python3 -m venv /path/to/venv
```

Например:

```
sudo python3 -m venv /usr/local/venv
```

5. Выдайте права на папку для venv

```
sudo chmod -R a+rwX /path/to/venv
```

6. Активируйте venv

```
source /path/to/venv/bin/activate
```

7. Установите пакеты

numpy

```
python3 -m pip install /path/to/prerequisitesFolder/python3/Debian\ 12/numpy-2.1.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

PyYAML

```
python3 -m pip install /path/to/prerequisitesFolder/python3/Debian\ 12/PyYAML-6.0.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

ctranslate2

```
python3 -m pip install /path/to/prerequisitesFolder/python3/Debian\ 12/ctranslate2-2.23.0-cp311-cp311-linux_x86_64.whl
```



/path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

8. Выйдите из venv

```
deactivate
```

Astra Linux 1.7 SE

Python 3.7 включен в дистрибутив Astra Linux 1.7 SE, в связи с чем установка самого Python на данной ОС не требуется. Необходимо установить pip и пакеты для Python.

Для установки необходимых компонентов выполните следующую последовательность действий:

1. Установите lib2to3 и distutils

```
sudo dpkg -i /path/to/prerequisitesFolder/python3/Astra\ SE\ 1.7/python3-  
lib2to3_3.7.3-1+b3_all.deb  
sudo dpkg -i /path/to/prerequisitesFolder/python3/Astra\ SE\ 1.7/python3-  
distutils_3.7.3-1+b3_all.deb
```

 /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

2. Установите pip

```
sudo python3 /path/to/prerequisitesFolder/python3/pip-23.0.1-py3-none-any.whl/pip  
install /path/to/prerequisitesFolder/python3/pip-23.0.1-py3-none-any.whl
```

 /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

3. Установите необходимые пакеты

numpy

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/Astra\ SE\  
1.7/numpy-1.21.6-cp37-cp37m-manylinux_2_12_x86_64.manylinux2010_x86_64.whl
```

PyYAML

```
sudo python3 -m pip install --ignore-installed  
/path/to/prerequisitesFolder/python3/Astra\ SE\ 1.7/PyYAML-6.0-cp37-cp37m-  
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.  
whl
```

ctranslate2

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/Astra\ SE\  
1.7/ctranslate2-2.14.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

 /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

Astra Linux 1.8 SE

Python 3.11.2 включен в дистрибутив AstraLinux 1.8 SE, в связи с чем установка самого Python на данной ОС не требуется. Необходимо установить venv и пакеты для Python.

 При установке набора на Astra Linux 1.8 SE необходимо указывать python3 из виртуального окружения /path/to/venv/bin/python3, в противном случае установленные пакеты находиться не будут.

Для установки необходимых компонентов выполните следующую последовательность действий:

1. Обновите список пакетов

```
sudo apt update
```

2. Установите venv

```
sudo apt install python3-venv
```

3. Создайте папку для venv

```
sudo mkdir /path/to/venv
```



/path/to/venv – путь к папке, в которой будет развернут venv

4. Создайте venv

```
sudo python3 -m venv /path/to/venv
```

Например:

```
sudo python3 -m venv /usr/local/venv
```

5. Выдайте права на папку для venv

```
sudo chmod -R a+rwX /path/to/venv
```

6. Активируйте venv

```
source /path/to/venv/bin/activate
```

7. Установите пакеты

numpy

```
python3 -m pip install /path/to/prerequisitesFolder/python3/Astra\ SE\ 1.8/numpy-2.1.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

PyYAML

```
python3 -m pip install /path/to/prerequisitesFolder/python3/Astra\ SE\ 1.8/PyYAML-6.0.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

ctranslate2

```
python3 -m pip install /path/to/prerequisitesFolder/python3/Astra\ SE\ 1.8/ctranslate2-2.23.0-cp311-cp311-linux_x86_64.whl
```



/path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

8. Выйдите из venv

```
deactivate
```

ALT Linux 10.0

По умолчанию в AltLinux 10.0 установлен python 3.9.6, который можно использовать для работы с PAI, и дополнительных действий, кроме установки pip и пакетов, не требуется.

Для установки необходимых компонентов выполните следующую последовательность действий:

1. Установите pip

```
sudo python3 /path/to/prerequisitesFolder/python3/pip-23.0.1-py3-none-any.whl/pip
install /path/to/prerequisitesFolder/python3/pip-23.0.1-py3-none-any.whl
```

 /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

2. Установите необходимые пакеты:

numpy

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/ALT\ Linux\
10.0/numpy-1.21.6-cp39-cp39-manylinux_2_12_x86_64.manylinux2010_x86_64.whl
```

PyYAML

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/ALT\ Linux\
10.0/PyYAML-6.0-cp39-cp39-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.
whl
```

ctranslate2

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/ALT\ Linux\
10.0/ctranslate2-2.14.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

 /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

РЕД ОС 7.3

По умолчанию в РЕД ОС 7.3 установлен Python версии 3.8.2, который можно использовать для работы с PAI, и дополнительных действий, кроме обновления pip и установки пакетов, не требуется.

Для установки необходимых компонентов выполните следующую последовательность действий:

1. Обновите pip

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/pip-23.0.1-py3-
none-any.whl
```

 /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

2. Установите пакеты:

numpy

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/RED\ OS\ 7/numpy-
1.24.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

PyYAML

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/RED\ OS\
7/PyYAML-6.0-cp38-cp38-
```

```
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.whl
```

ctranslate2

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/RED\ OS\ 7/ctranslate2-2.14.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

 /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

РЕД ОС 8

По умолчанию в РЕД ОС 8 установлен Python версии 3.11, который можно использовать для работы с PAI.

Для установки необходимых компонентов выполните следующую последовательность действий:

1. Проверьте наличие pip:

```
sudo python3 -m pip --version
```

- 1.1. Если pip не установлен, установите его:

```
sudo python3 /path/to/prerequisitesFolder/python3/pip-23.0.1-py3-none-any.whl/pip install /path/to/prerequisitesFolder/python3/pip-23.0.1-py3-none-any.whl
```

 /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.

- 1.2. Если pip уже установлен, то никаких дополнительных действий, кроме установки пакетов, не требуется

2. Установите необходимые пакеты:

Пакет PyYAML (версия 6.0.1) установлен по умолчанию, поэтому нужно установить только пакеты numpy и ctranslate2:

numpy

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/RED\ OS\ 8/numpy-2.1.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

ctranslate2

```
sudo python3 -m pip install /path/to/prerequisitesFolder/python3/RED\ OS\ 8/ctranslate2-2.23.0-cp311-cp311-linux_x86_64.whl
```

 /path/to/prerequisitesFolder – путь к папке Prerequisites после распаковки архива с дистрибутивом программы.