

**ДОКУМЕНТАЦИЯ, СОДЕРЖАЩАЯ ОПИСАНИЕ  
ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК  
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**PROMT Neural Translation Server – Training Addon  
(для ОС Linux)**

**И ИНФОРМАЦИЮ, НЕОБХОДИМУЮ ДЛЯ  
УСТАНОВКИ И ЭКСПЛУАТАЦИИ ПРОГРАММНОГО  
ОБЕСПЕЧЕНИЯ**

**PROMT Neural Translation Server  
(Linux) –  
Training Addon 24**

**Руководство администратора**

# PROMT Neural Translation Server (Linux) – Training Addon 24

## Руководство администратора

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

© 2003–2023, ООО «ПРОМТ». Все права защищены.

Россия, 199155,

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

E-mail: [common@promt.ru](mailto:common@promt.ru)

[support@promt.ru](mailto: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 Neural Translation Server (Linux) – Training Addon .....	6
2.2. Помощь и сопровождение .....	6
3. Установка.....	8
3.1. Системные требования .....	8
3.1.1. Требования к серверной части.....	8
3.1.2. Требования к клиентской части.....	9
3.2. Установка продукта .....	10
3.3. Активация PNTS Linux Training Addon .....	12
3.4. Настройки конфигурационного файла appsettings.json.....	13
3.4.1. Настройка доступа по https-протоколу .....	13
3.4.2. Задание индекса используемого GPU.....	14
3.4.3. Задание дополнительных параметров запуска утилиты обучения .....	14
3.5. Веб-интерфейс PNTS Linux Training Addon.....	16
4. Общая схема работы.....	18
5. Подготовка исходных данных.....	19
5.1. Текстовый корпус .....	19
6. Настройка модели .....	21
6.1. Задание параметров и запуск процесса настройки .....	21
6.2. Процесс настройки.....	24
6.2.1. Этапы обучения модели .....	25
6.2.1.1. Препроцессинг .....	25
6.2.1.2. Обучение модели.....	27
6.3. Результат обучения.....	28
7. Загрузка настроенной модели на сервер .....	30
8. Оценка результатов .....	32
Глоссарий.....	35
Приложение 1. Префиксы для обозначения языков .....	37
Приложение 2. Установка Python на системах Linux .....	39
Ubuntu 20.04/22.04.....	39
Ubuntu 23.04.....	39
Debian 10/11 .....	40
Astra Linux 1.7 SE.....	40
ALT Linux 10.0.....	41
РЕД ОС 7.3.....	42


# 1. О документации

## 1.1. Состав документации

Документация *PROMT Neural Translation Server (Linux) – Training Addon* содержит сведения об установке указанного компонента, а также информацию, необходимую для настройки (обучения) моделей нейронного машинного перевода (NMT-моделей), используемых в переводчике PROMT, с помощью корпуса параллельных двуязычных текстов.





 Данное руководство предназначено для пользователей переводчика *PROMT Neural Translation Server* (включая его модификации), осуществляющих настройку NMT-моделей посредством дополнительной функциональности переводчика, реализованной в компоненте *PROMT Neural Translation Server (Linux) – Training Addon*.

 *PROMT Neural Translation Server (PNTS)* — это единый набор программных модулей, позволяющих получить нейронный машинный перевод через веб-интерфейс.

 Подробные сведения о работе с переводчиком *PNTS* см. в руководстве пользователя или в руководстве администратора *PNTS*.

## 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 PROMT Neural Translation Server используется нейронный машинный перевод (Neural Machine Translation — NMT) — разновидность машинного перевода, в основе которого лежит механизм двунаправленных рекуррентных нейронных сетей (Bidirectional Recurrent Neural Networks), построенный на матричных вычислениях. 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 Neural Translation Server (Linux) – Training Addon

*PROMT Neural Translation Server (Linux) – Training Addon* (далее *PNTA*) является дополнением для программного продукта *PROMT Neural Translation Server* (далее *PNTS*), включая его модификации, и предназначен для изменения рабочих характеристик нейронного машинного перевода в программном продукте за счет автоматизированной настройки (обучения, тюнинга) моделей нейронного машинного перевода (NMT-моделей). Автоматизированная настройка производится с помощью корпуса параллельных двуязычных текстов, предоставляемых пользователем.

Компонент *PNTS Linux Training Addon* расширяет функциональность установленного у пользователя программного продукта *PNTS* за счет дополнительной функциональной возможности автоматической настройки моделей нейронного машинного перевода.

Входными данными для системы настройки являются NMT-модель (в виде zip-архива) и корпус параллельных текстов.

Результатом работы является настроенная по текстовому корпусу новая модель (в виде zip-архива), которую необходимо загрузить в *PNTS*.

### 2.2. Помощь и сопровождение

В случае возникновения проблем при работе с программным продуктом 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 видекарты 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 рядом с компьютером.

## 3. Установка

Система устанавливается с помощью инсталляционного набора.

 Установка должна осуществляться с правами суперпользователя.

### 3.1. Системные требования

#### 3.1.1. Требования к серверной части

- Выделенный сервер (физический или виртуальный) 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 и с поддержкой CUDA 12\*
- Оперативная память: 32 Гб (при объеме тренировочных данных до 1 млн. сегментов)

Место на диске: 20 Гб

\* требуется установить последнюю версию драйвера GPU для соответствующей ОС. Чем больший объем памяти будет выделен, тем выше будет результат обучения модели.

Поддерживается работа на следующих ОС:

- Ubuntu 20.04
- Ubuntu 22.04
- Ubuntu 23.04
- AstraLinux SE 1.7
- Debian 10
- Debian 11
- РЕД ОС 7.3
- ALT Linux 10.0

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 3.10 включительно
- пакет Numpy для Python
- пакет PyYAML 6.0 для Python
- пакет CTranslate2 2.14.0 для Python

Программное обеспечение mono, которое используется при работе *PNTS Linux Training Addon*, также входит в состав дистрибутива.

### **3.1.2. Требования к клиентской части**

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

- Chrome
- Firefox
- Opera
- Edge
- Safari (macOS)
- Яндекс-браузер

## 3.2. Установка продукта

*PNTS Linux Training Addon* функционирует совместно с программным продуктом *PNTS*. Для использования моделей, обученных с помощью *PNTA*, требуется наличие у пользователя *PNTS* с действующей лицензией.

Запустите установочный файл, используя следующую команду:

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

где *PNTS\_Training\_Addon\_Uxxxx.run* - название установочного файла *PNTS Linux Training Addon* (xxxx – номер версии).

Например, если установочный файл называется *PNTS\_Training\_Addon\_U0017.run*:

```
sudo chmod +x PNTS_Training_Addon_U0017.run && sudo ./PNTS_Training_Addon_U0017.run
```

Процесс установки состоит из нескольких шагов, которые выполняются автоматически.

В процессе установки:

1. Ознакомьтесь с лицензионным соглашением и примите его условия.

 При отказе от условий лицензионного соглашения дальнейшая установка невозможна.

2. При необходимости укажите папку, в которую будет установлен программный продукт. По умолчанию *PNTS Training Addon* устанавливается в папку */usr/local/pntaXX*.

Например: */usr/local/pnta24*

3. Программные модули и лингвистические данные распаковываются в корневую папку продукта. Происходит настройка исполняемых файлов и установка разрешений.

4. При необходимости укажите адрес, по которому расположен веб-интерфейс *PNTS*. Адрес по умолчанию *http://127.0.0.1/pnts*

5. Настройка фаервола.

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

6. Задайте имя веб-приложения.

По умолчанию имя веб-приложения – *pnta*

7. Укажите пользователя сервисов *PNTA*

По умолчанию пользователь - *pnta*

8. Укажите версию *python*.

По умолчанию – *python3*

 Следует указывать именно ту версию *python*, с которой предстоит работать.

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

Сразу после установки на систему РЕД ОС 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/pnta24/addon/dotnet/dotnet`

Измените метку каталога с помощью следующей команды:

```
sudo /sbin/restorecon -v /usr/local/pnta24/addon/dotnet/dotnet
```

**!** В случае игнорирования данных уведомлений безопасности SELinux при установке на РЕД ОС веб-интерфейс *PNTS Linux Training Addon* будет недоступен.

### 3.3. Активация PNTS Linux Training Addon

В продукте реализована система лицензирования, позволяющая ограничивать доступ к основной функциональности продукта в зависимости от файла лицензии.

Вместе с набором устанавливается лицензионный файл *license.lic*, располагающийся в корневой папке набора.

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/pnta24/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. Перезапустите сервис `pnta`.

```
sudo systemctl restart promt.trainingaddon.service
```

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

 Если продукт не активирован, корректная работа аддона невозможна.

## 3.4. Настройки конфигурационного файла `appsettings.json`

### 3.4.1. Настройка доступа по https-протоколу

По умолчанию *PNTA* устанавливается с доступом только по http-протоколу.

В случае необходимости администратор может настроить доступ по https-протоколу. Для этого необходимо добавить блок "Https" в конфигурационный файл `appsettings.json` (по умолчанию `/usr/local/pnta24/addon/app/appsettings.json`), прописав в нём следующие параметры:

<i>Url</i>	url, на котором будет работать PNTA
<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>	true

Пример секции файла конфигурации до настройки:

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

Пример настройки:

```
"Endpoints": {  
  "Http": {  
    "Url": "http://+:80"  
  },  
  "Https": {  
    "Url": "https://+:443",  
    "Certificate": {  
      "Path": "/home/user/pserver.pem",  
      "KeyPath": "/home/user/pserver_converted.key",  
      "Password": "qwe123",  
      "AllowInvalid": true  
    }  
  }  
}
```

```
}  
}  
}
```

### 3.4.2. Задание индекса используемого GPU

Если предполагается работать на системе с несколькими GPU, то чтобы выбрать определенное устройство, необходимо прописать его индекс в параметре *Model\_GPU\_Devices* конфигурационного файла *appsettings.json* (по умолчанию *pnta24/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 аддона. Пример вызова утилиты при установке в папку по умолчанию:

```
/usr/local/pnta24/utils/promt-gpuinfo.exe
```

Первое выведенное в результатах устройство имеет индекс "0".

Пример настройки:

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

Если указанный идентификатор выходит за границы диапазона, процесс обучения будет остановлен с сообщением об ошибке.

### 3.4.3. Задание дополнительных параметров запуска утилиты обучения

Основная настройка перед запуском обучения осуществляется через веб-интерфейс *PNTA* на странице *Данные для обучения* группы страниц *Построение модели* (подробнее см. [Задание параметров и запуск процесса настройки](#)). В случае необходимости вы можете дополнительно задать некоторые параметры для запуска утилиты обучения в конфигурационном файле *appsettings.json* (по умолчанию *pnta24/addon/app/appsettings.json*). Параметры могут быть заданы в разделе *TrainingUtility/DefaultSettings*.

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>python_path</i>	Путь для вызова python нужной версии, настраивается при установке аддона. По умолчанию – python3. Версия Python должна соответствовать системным требованиям и для нее должны быть установлены соответствующие пакеты (см. раздел 3.1).
<i>CT2</i>	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. По умолчанию – 1 (конвертация включена).

### 3.5. Веб-интерфейс PNTS Linux Training Addon

Для проверки работы веб-интерфейса 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 – pnta.

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

🟡 Откроется страница веб-интерфейса:

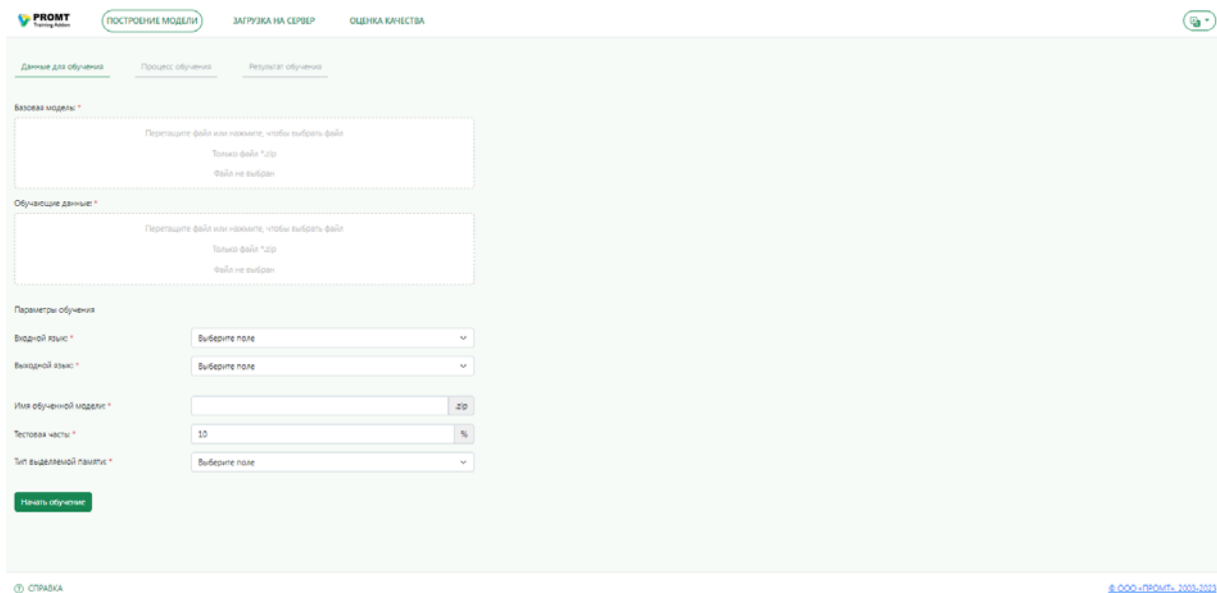


Рис. 3.1 Общий вид веб-интерфейса PNTA

Главное меню *PROMT Training Addon* представлено в виде заголовков в верхней части страницы:

- *Построение модели*

Группа страниц *Построение модели* содержит несколько страниц, обеспечивающих доступ к функциональности аддона:

- *Данные для обучения*

Для загрузки данных для обучения, настройки параметров обработки, задания имени результирующей модели и запуска обучения.



— *Процесс обучения*

Для контроля процесса обучения модели.

— *Результат обучения*

Доступны следующие команды:


- *Скачать обученную модель для загрузки на сервер перевода*
- *Скачать текстовую часть для оценки качества перевода*
- *Скачать лог обучения для анализа процесса обучения*
- *Начать новую сессию обучения*

• *Загрузка на сервер*

Для загрузки моделей на сервер *PNTS*.

• *Оценка качества*

Для оценки результатов настройки.


Справка к продукту доступна по нажатию на кнопку  *Справка* в нижнем левом углу страницы приложения.

## 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

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

### 4. Загрузка настроенной модели на сервер

- Установите настроенную NMT-модель в *PNTS* 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

### 5.1. Текстовый корпус

Исходные данные (текстовый корпус) должны быть представлены в виде архива формата zip. Содержимое архива – это один или несколько tmx файлов 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 имелась пара файлов с одним и тем же названием, но имеющих разные расширения, соответствующие префиксам входного и выходного языков.

 Языковые префиксы см. в [Приложении 1](#).

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.[Stgt]	dev.[Src]
dev.[Stgt]	dev.tok.[Src]
dev.tok.[Stgt]	test.[Src]
test.[Stgt]	

Таблица 5.1 Зарезервированные имена файлов

**Примеры зарезервированных имен 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:**

corpus.tmx  
corpus1.en  
corpus1.ru  
corpus2.en  
corpus2.ru

## 6. Настройка модели

В качестве исходных данных для настройки модели используется архив базовой модели. Прежде чем приступить к настройке, скопируйте архив модели, которую необходимо обучить, в любую папку.

 NMT-модели поставляются в виде архивов с расширением .zip.

### 6.1. Задание параметров и запуск процесса настройки

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

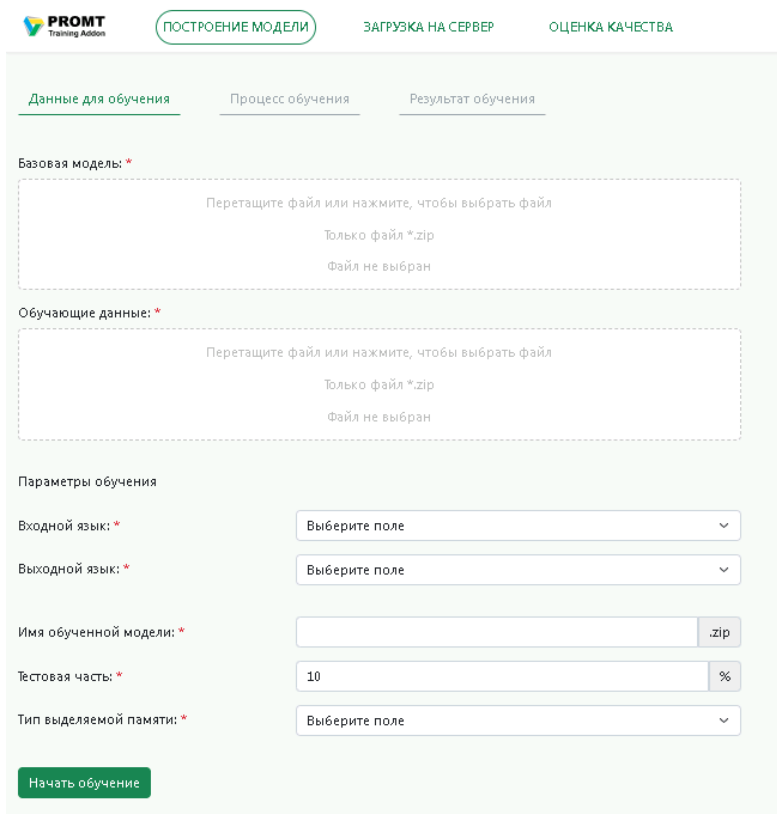


Рис. 6.1 Страница *Данные для обучения*

 Чтобы начать обучение модели, выполните указанную последовательность действий:

### 1. Выберите архив базовой модели

Щелкните мышью в поле *Базовая модель* и выберите необходимый файл или перетащите файл базовой модели в данное поле.

✿ В поле *Базовая модель* может быть загружен только файл формата .zip.

✿ Базовая модель для обучения входит в комплект поставки PNTA.

### 2. Выберите архив с обучающими данными

Щелкните мышью в поле *Обучающие данные* и выберите необходимый файл или перетащите файл архива данных в данное поле.

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

### 3. Задайте параметры обучения:

- В строке *Входной язык* выберите необходимый язык из списка доступных языков. Аналогичным образом выберите необходимый язык из списка в строке *Выходной язык*.

✿ Список доступных языков ограничивается файлом лицензии.

- Задайте имя для настроенной модели, введя его в поле *Имя обученной модели*.

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

- Укажите суммарный объем корпуса для валидации и тестового корпуса в процентах от исходного корпуса в строке *Тестовая часть*.

✿ Рекомендуется задавать данный параметр в пределах от 1 до 10%. Суммарное максимальное число строк, выделяемое для двух этих корпусов, не может превышать 6000. При этом исходный обучающий корпус уменьшается на соответствующее количество строк.

- В строке *Тип выделяемой памяти* выберите зарезервированный размер памяти, отведенный под процесс настройки модели:

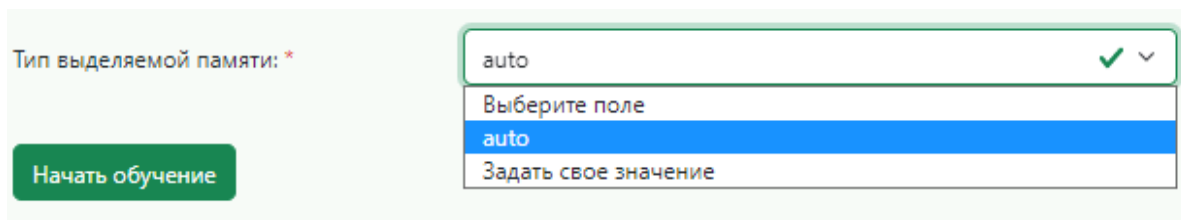


Рис. 6.2 Выбор типа выделяемой памяти

Если в качестве значения указано *auto*, то размер выделяемой памяти вычисляется перед запуском настройки модели, исходя из доступной памяти GPU-устройства и из размера базовой модели.

Чтобы указать конкретное значение, выберите *Задать свое значение* и укажите объем памяти в мегабайтах в строке *Выделение памяти*:

Параметры обучения

Входной язык: *	Английский	✓	▼
Выходной язык: *	Русский	✓	▼
Имя обученной модели: *	ER_trained	✓	.zip
Тестовая часть: *	9	✓	%
Тип выделяемой памяти: *	Задать свое значение	✓	▼
Выделение памяти: *	2000	✓	M6

**Начать обучение**

Рис. 6.3 Задание значения для выделения памяти

💡 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.

4. Нажмите кнопку *Начать обучение*, чтобы запустить процесс настройки модели.

🔹 Если все настройки были указаны верно, будет произведена загрузка данных на сервер и запущен процесс обучения модели.

## 6.2. Процесс настройки

После старта процесса настройки в окне на странице *Процесс обучения* выводится лог процесса обучения модели:



```
Start training
CORPUS=/usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4
BASELINE_MODEL=/usr/local/pnta24/addon/storage/training/base_models/9eaf57be-02ce-4dc0-83ce-0d5c67597dd/tempModel/re/vr_24_0_07_23.zip
BASE_CORPUS=/usr/local/pnta24/addon/storage/training/base_models/9eaf57be-02ce-4dc0-83ce-0d5c67597dd/tempModel/general
0) prompt.nts.extractcorpus.exe:
/usr/local/pnta24/utills/prompt.nts.extractcorpus.exe /src:ru /tgt:en /p:"/usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4"
/usr/local/pnta24/utills/prompt.nts.marlablauncher.exe --prepare-model --pretrained-model "/usr/local/pnta24/addon/storage/training/base_models/9eaf57be-02ce-4dc0-83ce-0d5c67597dd/tempModel/re/vr_24_0_07_23.zip" --model
"/usr/local/pnta24/addon/storage/training/results/rn_new_test.zip"
MarlabLauncher
Calculating dim-vocabs:
46981 34299
/usr/local/pnta24/addon/storage/training/base_models/9eaf57be-02ce-4dc0-83ce-0d5c67597dd/tempModel/model/re/vr_24_0_07_23.zip
/usr/local/pnta24/utills/prompt.nts.tokenize.exe --pt --lang ru < /usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4/corpus.ru > /usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4/corpus.tok.def.ru
4) prompt.nts.tokenize.exe:
/usr/local/pnta24/utills/prompt.nts.tokenize.exe --pt --lang en < /usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4/corpus.en > /usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4/corpus.tok.def.en
5) prompt.nts.cleancorpus.exe:
/usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4/corpus.tok.def.ru" -it "/usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4/corpus.tok.def.en" --dedup
Rename: /usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4/corpus.tok.clean.ru /usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4/corpus.tok.ru
6) prompt.nts.extractrandomset.exe:
/usr/local/pnta24/utills/prompt.nts.extractrandomset.exe -p:"/usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4" -corpus:corpus.tok -src:ru -tgt:en -res:"dev.tok test.tok" -percent:10 -max:6000
9) prompt.nts.tokenize.exe:
/usr/local/pnta24/utills/prompt.nts.tokenize.exe --pt --detok --lang ru < /usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4/test.tok.ru > /usr/local/pnta24/addon/storage/training/corpuses/ffde4c23-3e8b-4213-901a-d436d8868af4/test.ru
```

Рис. 6.4 Вкладка *Процесс обучения*

В случае возникновения ошибки после запуска процесса `marlab.exe`, нажмите кнопку *Скачать журнал* или скопируйте содержимое окна с логом процесса обучения.

- Чтобы прервать обучение модели, нажмите на кнопку *Остановить обучение*.



## 6.2.1. Этапы обучения модели

Весь процесс обучения модели можно разбить на два этапа: препроцессинг текстового корпуса и непосредственно обучение модели.

### 6.2.1.1. Препроцессинг

Препроцессинг состоит из следующих шагов:

#### 1. Создание исходного параллельного текстового корпуса по папке с входными файлами

Выполняется утилитой *promt.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 *promt.nts.tokenize*. На вход подается корпус *corpus*, на выходе получается токенизированный корпус *corpus.tok*.

#### 3. Прочистка пользовательского корпуса

Производится утилитой *promt.nts.cleancorpus*. 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.

Кроме этого, утилита *promt.nts.cleancorpus.exe* в процессе работы 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
<i>Duplicate</i>	Пара вход-выход уже встречалась ранее
<i>Equal lines</i>	Вход = выход
<i>Bad length</i>	Вход или выход имеет меньше 1 слова (пустой), больше 100 слов, либо менее 10 символов

<i>Bad alphanumeric % in source/target</i>	Во входной/ выходной строке кол-во букв < 30% от всего входа
<i>Source/Target is empty</i>	После выполнения обработки во входной/выходной строке не осталось символов
<i>Source or target is too long or too short</i>	После выполнения обработки во входной/выходной строке стало меньше 1 или больше 100 слов
<i>Bad source length/target length ratio</i>	После выполнения обработки отношение длин входного текста к выходному > 9 или наоборот, если отношение выходного к входному >9.

#### 4. Создание корпуса для валидации данных и тестового корпуса

Выполняется утилитой *promt.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-set) и тестовым корпусом (test-set). 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.

#### 5. Генерация синтетических данных с незнакомыми словами для пользовательского корпуса

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

#### 6. Токенизация тестовых, валидационных и синтетических данных

Производится утилитой *promt.nts.tokenize*. На выходе получают обработанные файлы основного сета, синтетического сета и сета dev с дополнительной строкой .bpe. в названии.

#### 7. Построение выравнивания для пользовательских данных

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

#### 8. Подсчет количества необходимых итераций для обучения.

Осуществляется по формуле  $1000000/\text{кол-во пользовательских данных} + 1$ .

### 6.2.1.2. Обучение модели

После подготовки и обработки исходных данных запускается непосредственно процесс обучения (настройки) NMT-модели с помощью приложения командной строки **marian.exe**.

Непосредственно запуск **marian.exe** выполняется утилитой *promt.nts.MarianLauncher.exe*.

Утилита **promt.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.

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 "alignment: 0", переименовывает в "model.baseline.npz.decoder.yml".

3. Вычисляет значения параметра dim-vocab, анализируя файлы vocab.src.yml и vocab.tgt.yml;

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 *Result\_Model*.

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

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

### 6.3. Результат обучения

После окончания процесса обучения пользователю становится доступна страница *Результат обучения*.

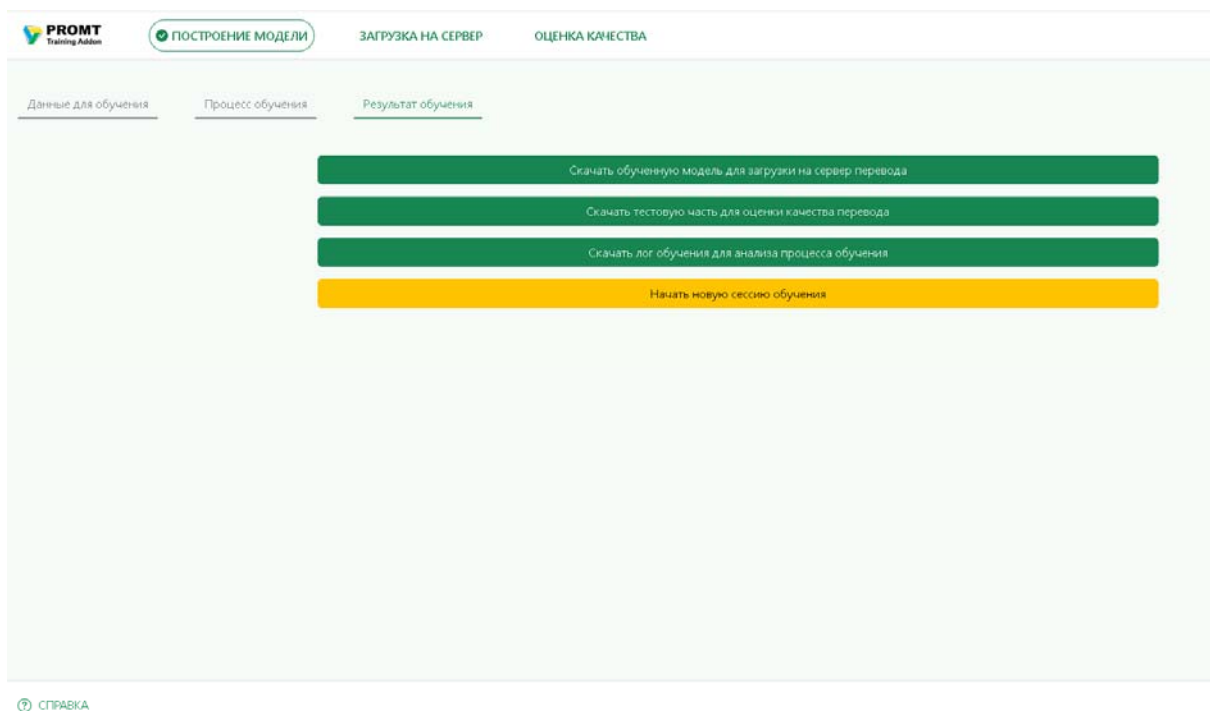



Рис. 6.4 Вкладка *Результат обучения*

Используйте функции окна на странице *Результат обучения* следующим образом:

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

 Модель скачивается в виде файла zip. Скачанная таким образом модель может быть затем загружена в *PNTS* (см. [Загрузка настроенной модели на сервер](#)).

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

✿ Текстовая часть скачивается 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 скачивается в виде архива log\_data.zip. Архив содержит логи <modelname>\_marian.train.log, <modelname>\_marian.valid.log и corpus.tok.def.clean.badstring.txt.

- Чтобы запустить новое обучение, выберите кнопку *Начать новую сессию обучения*.

✿ Перед началом новой сессии следует сохранить все необходимые данные, воспользовавшись кнопками на странице *Результат обучения*. В случае если настроенная модель не была сохранена по кнопке *Скачать обученную модель для загрузки на сервер перевода*, вы можете найти её в папке продукта pnta/addon/storage.

## 7. Загрузка настроенной модели на сервер

Обученная модель может быть выгружена из *PNTA* и затем загружена в *PNTS*.

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

ПROMT Training Addon

ПОСТРОЕНИЕ МОДЕЛИ

ЗАГРУЗКА НА СЕРВЕР

ОЦЕНКА КАЧЕСТВА

Файл модели: \*

Перетащите файл или нажмите, чтобы выбрать файл

Только файл \*.zip

er\_for\_check\_4.zip

Входной язык: \* Английский ✓

Выходной язык: \* Русский ✓

Название профиля: \* er\_from\_pnta\_to\_ubuntu ✓

Адрес сервера: \* http://127.0.0.1/pnts

Имя пользователя:

Пароль:

Загрузить модель на сервер

Рис. 7.1 Страница загрузки модели на сервер

► Чтобы загрузить настроенную модель в *PNTS*, выполните указанную последовательность действий:


### 1. Выберите файл с архивом настроенной модели

Щелкните мышью в поле *Файл модели* и выберите необходимый файл или перетащите файл с архивом настроенной модели в данное поле.


### 2. Выберите входной язык


3. Выберите выходной язык


4. 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.

5. Нажмите кнопку *Загрузить модель на сервер*.

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

 В поле *Адрес сервера* прописан адрес, указанный при установке аддона. Заменить адрес сервера можно только в файле конфигурации web-приложения *pnta24/addon/app/appsettings.json* в секции "*PntsProxy*". После уточнения файла конфигурации необходимо перезапустить сервис pnta (см. [Активация PNTS Linux Training Addon](#)).

 Начнется процесс загрузки модели на сервер.

Если после начала загрузки выводится сообщение о сбое при загрузке модели на сервер, проверьте логи аддона PNTA (по умолчанию логи находятся в папке `/usr/local/pnta24/addon/logs`) и логи PNTS (см. Руководство администратора PNTS).

## 8. Оценка результатов

Качество обученной модели оценивается с помощью вычисленного значения метрики BLEU, которое можно получить на странице *Оценка качества*.

The screenshot shows the 'Оценка качества' (Quality Evaluation) page in the PROMT Training Addon interface. The page has a light green background and a top navigation bar with 'Построение модели' and 'Загрузка на сервер' buttons. The main content area contains two file upload sections: 'Файл для перевода' (File for translation) and 'Файл эталонного перевода' (Reference translation file). Below these are dropdown menus for 'Входной язык' (English), 'Выходной язык' (Russian), 'Профиль (базовая модель)' (Universal), and 'Профиль (обученная модель)' (Business). There are also input fields for 'Адрес сервера' (http://127.0.0.1/pnits), 'Имя пользователя', and 'Пароль'. A green button 'Посчитать метрику BLEU' is at the bottom. A 'Справка' (Help) link is at the bottom left.

Рис. 8.1 Страница *Оценка качества*

▶ Для оценки результатов настройки модели выполните следующее:

### 1. Выберите 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 может быть загружен только файл test.\$src.

2. Выберите 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 может быть загружен только файл test.\$trg.


3. Выберите входной язык


4. Выберите выходной язык


5. Выберите профиль с базовой моделью

6. Выберите профиль с настроенной моделью

7. Нажмите кнопку Посчитать метрику BLEU

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

 В поле *Адрес сервера* прописан адрес, указанный при установке аддона. Заменить адрес сервера можно только в файле конфигурации web-приложения *pnta24/addon/app/appsettings.json* в секции "*PntsProxy*". После уточнения файла конфигурации необходимо перезапустить сервис pnta (см. [Активация PNTS Linux Training Addon](#)).

 Начнется процесс вычисления BLEU.

Будет выполнена обработка файла для перевода с базовым профилем и с профилем с настроенной моделью. Затем будет вычислено значение BLEU для базовой модели и значение BLEU для настроенной модели.

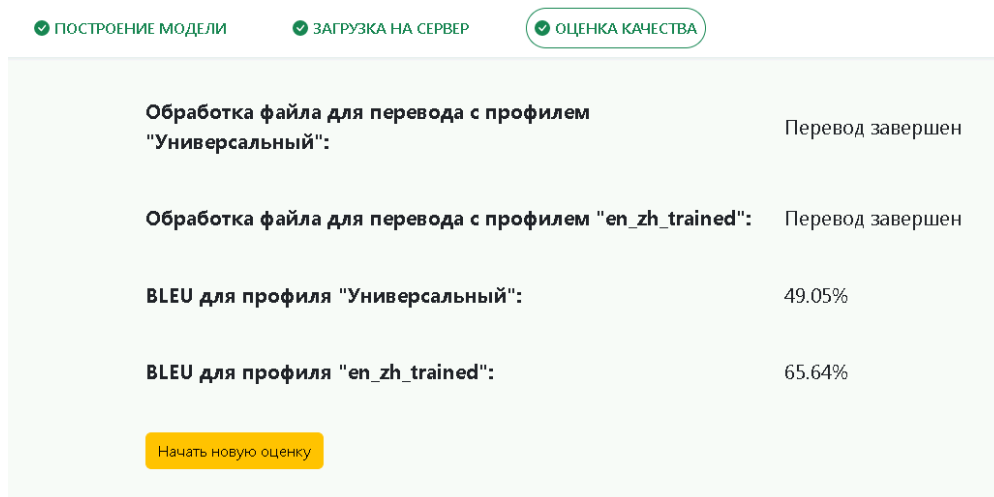


Рис. 8.1 Результаты BLEU на странице Оценка качества

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

## 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.

**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). Из входного корпуса выбранные строки удаляются. 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, принимается решение об окончании обучения.

**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). Из входного корпуса выбранные строки удаляются. 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 делается 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** – знаки пунктуации приводятся к последовательности `&`; (экранирование).

**Построение выравнивания** – токены из входной строки соотносятся с токенами из выходной строки, и результат записывается 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.

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

**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.

**Эпоха** – этап обучения, при котором единожды просматривается весь обучающий корпус просмотр во время обучения всего корпуса.

**Языковая пара** – указывает, с какого языка и на какой будет переводиться текст.

## Приложение 1. Префиксы для обозначения языков

В PNTS для обозначения языков используются префиксы в формате RFC. В таблице ниже приведены префиксы языков, которые используются при запуске обучения моделей, а также 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.

Язык	префикс
Азербайджанский	az
Английский	en
Арабский	ar
Армянский	hy
Белорусский	be
Болгарский	bg
Венгерский	hu
Вьетнамский	vi
Греческий	el
Грузинский	ka
Датский	da
Иврит	he
Индонезийский	id
Испанский	es
Итальянский	it
Казахский	kk
Каталанский	ca
Киргизский	ky
Китайский (традиционный)	zh-tw
Китайский (упрощенный)	zh-cn
Корейский	ko
Латышский	lv
Литовский	lt
Малайский	ms
Мальтийский	mt
Немецкий	de

Нидерландский	nl
Норвежский	nb
Нюнорск	nn
Польский	pl
Португальский	pt
Румынский	ro
Русский	ru
Сербский	sr
Словацкий	sk
Словенский	sl
Таджикский	tg
Тайский	th
Тамильский	ta
Татарский	tt
Турецкий	tr
Туркменский	tk
Узбекский	uz
Украинский	uk
Урду	ur
Фарси	fa
Филиппинский	fil
Финский	fi
Французский	fr
Хинди	hi
Хорватский	hr
Чешский	cs
Чувашский	cv
Шведский	sv
Эстонский	et
Японский	ja

## Приложение 2. Установка Python на системах Linux

Для всех поддерживаемых систем, кроме AstraLinux 1.7 SE, ALT Linux и РЕД ОС, предусмотрена только онлайн-установка Python и необходимых пакетов.

Для AstraLinux 1.7 SE, ALT Linux и РЕД ОС установка необходимых компонентов может производиться как онлайн, так и из папки Prerequisites в составе набора.

### Ubuntu 20.04/22.04

Python включен в дистрибутив Ubuntu 20.02/22.04, поэтому установка самого Python на данных ОС не требуется. Необходимо установить pip и пакеты для Pyton.

Для этого:

1. Обновите список пакетов


```
sudo apt update
```

2. Установите pip

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

3. Установите пакеты

```
sudo python3 -m pip install ctranslate2==2.14.0
```

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

### Ubuntu 23.04

Необходимо установить Python 3.10, pip и пакеты для Pyton.

1. Добавьте репозитории

```
sudo add-apt-repository --yes deb http://security.ubuntu.com/ubuntu jammy-security main
```

```
sudo add-apt-repository --yes deb-src http://security.ubuntu.com/ubuntu jammy-security main
```

```
sudo add-apt-repository --yes deb http://security.ubuntu.com/ubuntu jammy main
```

2. Установите python


```
sudo apt install python3.10
```

3. Установите pip

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

#### 4. Установите пакеты

```
sudo python3.10 -m pip install ctranslate2==2.14.0
```

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

## Debian 10/11

Python включен в дистрибутив Debian 10/11, поэтому установка самого Python на данных ОС не требуется. Необходимо обновить pip и установить пакеты для Python.

#### 1. Обновите pip

```
sudo python3 -m pip install --upgrade pip
```

#### 2. Установите пакеты

```
sudo python3 -m pip install ctranslate2==2.14.0
```

## Astra Linux 1.7 SE

Python3.7 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. Установка может производиться как онлайн, так и из папки Prerequisites в составе набора.

#### Онлайн-установка:

#### 1. Обновите список пакетов

```
sudo apt update
```

#### 2. Установите pip


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

#### 3. Обновите pip

```
sudo python3.7 -m pip install --upgrade pip
```

#### 4. Установите пакеты

```
sudo python3.7 -m pip install ctranslate2==2.14.0
```

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



### Офлайн-установка:

В случае работы в системе без интернет-подключения, установку необходимых компонентов можно осуществить в режиме офлайн.

Офлайн-установка выполняется из папки Prerequisites/python3 (по умолчанию /usr/local/pnta24/Prerequisites/python3).

Для установки необходимых компонентов выполните следующую последовательность действий:

#### 1. Установите pip

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

#### 2. Установите необходимые пакеты

- numpy

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

## **ALT Linux 10.0**

По умолчанию в Alt Linux 10.0 установлен python 3.9.6, который можно использовать для работы с *PNTA* и дополнительных действий, кроме установки pip, не требуется. Установка может производиться как онлайн, так и из папки Prerequisites/python3 в составе набора.

### Онлайн-установка:

#### 1. Обновите список пакетов

```
sudo apt-get update
```

#### 2. Установите pip

```
sudo apt-get install python3-module-pip
```

### Офлайн-установка:

### 1. Установите pip

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

### 2. Установите необходимые пакеты:

- numpy

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

## РЕД ОС 7.3

По умолчанию в RED OS 7.3 установлен Python 3.8.2, который можно использовать для работы с *PNTA*, и дополнительных действий, кроме обновления pip и установки пакетов, не требуется. Установка может производиться как онлайн, так и из папки *Prerequisites/python3* в составе набора.


### Онлайн-установка:

#### 1. Обновите pip

```
sudo python3 -m pip install --upgrade pip
```

#### 2. Установите пакеты

```
sudo python3 -m pip install ctranslate2==2.14.0
```

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

### Офлайн-установка:

Офлайн-установка выполняется из папки *Prerequisites/python3* (по умолчанию */usr/local/pnta24/Prerequisites/python3*).

Для установки необходимых компонентов выполните следующую последовательность действий:

#### 1. Обновите pip

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

## 2. Установите пакеты:

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

```
sudo python3 -m pip install ./RED\ OS/PyYAML-6.0-cp38-cp38-  
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.wh  
l
```

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