

ДОКУМЕНТАЦИЯ, СОДЕРЖАЩАЯ ОПИСАНИЕ  
ФУНКЦИОНАЛЬНЫХ ХАРАКТЕРИСТИК ПРОГРАММНОГО  
ОБЕСПЕЧЕНИЯ

PROMT Neural Translation Server – Training Addon  
(для ОС Windows)

И ИНФОРМАЦИЮ, НЕОБХОДИМУЮ ДЛЯ УСТАНОВКИ И  
ЭКСПЛУАТАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

# **PROMT Neural Translation Server – Training Addon 23**

**Руководство администратора**

# PROMT Neural Translation Server – Training Addon 23

## 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 <i>PROMT Neural Translation Server – Training Addon</i> .....	6
2.2. Помощь и сопровождение .....	6
3. Установка.....	8
3.1. Системные требования.....	8
3.2. Установка продукта.....	9
3.3. Активация продукта .....	9
4. Общая схема работы.....	11
5. Подготовка исходных данных .....	13
5.1. Текстовый корпус .....	13
5.2. Подготовка базовой модели .....	15
6. Настройка модели .....	16
6.1. Запуск процесса настройки и задание параметров.....	16
6.2. Этапы обучения модели.....	19
6.2.1. Препроцессинг.....	19
6.2.2. Обучение модели.....	21
7. Вычисление финального значения BLEU .....	22
7.1. Параметры для запуска утилиты.....	22
Глоссарий .....	24
Приложение 1. Префиксы для обозначения языков .....	26


# 1. О документации

## 1.1. Состав документации

Документация *PROMT Neural Translation Server – 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 (PNTS)*, включая его модификации, осуществляющих настройку NMT-моделей посредством дополнительной функциональности переводчика, реализованной в компоненте *PROMT Neural Translation Server – Training Addon*.

 PROMT Neural Translation Server — это масштабируемая клиент-серверная система машинного перевода, в которой для всех языковых пар используется нейронный перевод.

 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 пользователя.

## 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 – Training Addon*

*PROMT Neural Translation Server – Training Addon* (далее *PNTS Training Addon*) является дополнением для программного продукта PROMT Neural Translation Server (далее PNTS) и предназначен для изменения рабочих характеристик нейронного машинного перевода 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 Training Addon* расширяет функциональность установленного у пользователя программного продукта PNTS за счет дополнительной функциональной возможности автоматической настройки моделей нейронного машинного перевода.

*PNTS Training Addon* представляет собой набор утилит командной строки, которые последовательно запускаются с помощью bat-файла.

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 драйверов видеокарты/ видеокарт, версия CUDA;
- характеристики используемого ПО: версию ОС Windows, установленные пакеты обновления, локализацию и региональные установки;

- 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.

 Пользователь, который запускает процесс установки, должен обладать правами локального администратора.

### 3.1. Системные требования

В качестве ОС может использоваться десктопная или серверная система:

- Windows 10 x64, начиная с Windows10 Anniversary Update
- Windows 11
  
- Windows Server 2016
- Windows Server 2019
- Windows Server 2022

Минимальные аппаратные требования:

- Процессор Intel Core i5
- Видеокарта NVIDIA GTX 1070 (8 Гб) (рекомендуется NVIDIA GeForce GTX 1080 Ti)
- Оперативная память 16 Гб

Место на диске: 10 Гб.

Для работы инсталлятора требуется .NET Framework 3.5, который необходимо включить как компонент системы: Программы и компоненты > Включение или отключение компонентов Windows.

Для работы продукта требуется .NET Framework 4.8, входящий в состав набора.

Для работы утилиты конвертации моделей в формат CT2 PNTA поставляется с embedded-версией Python 3.7 и необходимыми пакетами.


#### Требования к GPU

Для обучения моделей используются вычислительные возможности видеокарты (GPU).

В качестве GPU могут использоваться видеокарты NVIDIA. Минимальная модель видеокарты: NVIDIA GTX 1070.

Минимальное требование к памяти: 8 Гб (рекомендуется 11 Гб и больше).




 Чем больший объем памяти будет выделен, тем выше будет результат обучения модели.

Требуется драйвер видеокарты с поддержкой CUDA 11.

## 3.2. Установка продукта

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

 Перед установкой завершите работу всех приложений Windows.


 В системе должны быть установлены последние обновления безопасности.

Запустите установочный файл, который представляет собой самораспаковывающийся архив с именем, например, *PNTS\_Training\_Addon.exe*. Нажмите кнопку *Далее*.


Процесс установки состоит из нескольких шагов, которые выполняются автоматически и прервать которые нельзя:

1. Распаковка архива.


2. Выдача стартового окна.

 Запускается *мастер установки*, который автоматически выполняет все необходимые действия.

3. Ознакомление с лицензионным соглашением.

 После ознакомления с лицензионным соглашением примите его условия или откажитесь. При отказе дальнейшая установка невозможна.

4. Выбор расположения папки установки продукта.

 При необходимости укажите папку, в которую будет установлен программный продукт. По умолчанию: *C:\Program Files (x86)\PROMT Neural Translation Server - Training Addon 23*.

5. Завершение установки.

## 3.3. Активация продукта

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. Данный лицензионный файл содержит информацию о статусе продукта (по умолчанию статус "Не активирован") и доступных языковых парах.

Чтобы активировать продукт, выполните следующее:

1. Получите идентификатор компьютера, привязанный к оборудованию системы, в которой установлен продукт, запустив bat-файл **hardwareid.bat**, находящийся в папке продукта (при установке по умолчанию *C:\Program Files (x86)\PROMT Neural Translation Server - Training Addon 23*)

 Запуск файла *hardwareid.bat* должен осуществляться из командной строки.

Результатом выполнения данного файла является строка с идентификатором **hardwareID**, которая выводится в консоль.

2. Пришлите в службу поддержки компании ПРОМТ лицензионный файл и идентификатор компьютера **hardwareID**.
3. Получив новый лицензионный файл, перепишите его "поверх" файла, установленного с набором.

Результатом активации является новый лицензионный файл, который привязан к оборудованию и содержит статус "Активирован".

Утилиты при запуске проверяют наличие и валидность лицензионного файла. Если продукт активирован, то утилиты работают без ограничений. Если продукт не активирован, корректная работа аддона невозможна.

## 4. Общая схема работы

*PNTS Training Addon* представляет собой набор утилит командной строки, которые последовательно запускаются с помощью bat-файла.

Файл **run\_training.bat** запускает препроцессинг и обучение модели. Для оценки результатов обучения модели используется файл **run\_bleuscore.bat**.

❗ Перед запуском bat-файлов необходимо задавать соответствующие параметры.

❗ Запуск bat-файлов должен осуществляться из командной строки.

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.

👉 Подробные сведения о подготовке исходных данных см. в разделе 5.

- Скопируйте модель, которую необходимо обучить, в любую папку.

❗ У пользователя, который запускает процесс настройки, должны быть права на запись в папку с моделью.

### 2. Настройка модели

- Задайте необходимые параметры в файле **utils\settings.json** и запустите файл **run\_training.bat**, который автоматически выполнит препроцессинг текстового корпуса и непосредственно обучение модели.

👉 Подробные сведения о запуске процесса настройки модели см. в разделе 6.1.

### 3. Сохранение результатов настройки модели

- Установите настроенную NMT-модель в *PNTS*.

👉 Подробные сведения об установке NMT-моделей в *PNTS* см. в Руководстве администратора *PNTS*.

### 4. Оценка результатов настройки модели с помощью значения BLEU


💡 Для оценки результатов настройки убедитесь, что базовая модель подключена к одному профилю перевода (например, "Универсальный"), а настроенная модель подключена к другому профилю (например, "test").

- a. Вычислите значение BLEU для базовой модели. Для этого задайте нужные параметры и запустите файл **run\_bleuscore.bat**.

- b. Вычислите значение BLEU для настроенной модели. Для этого задайте нужные параметры и запустите файл **run\_bleuscore.bat**.

с. Сравните полученные результаты BLEU из пп. а. и б. 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.

 Подробные сведения о параметрах запуска **run\_bleuscore.bat** см. в разделе 7.1.

 Подробную информацию об этапах настройки модели можно найти в следующих разделах данного документа.

## 5. Подготовка исходных данных

### 5.1. Текстовый корпус

Исходные данные (текстовый корпус) должны быть представлены либо в виде одного или нескольких tmx файлов, либо в виде нескольких текстовых файлов в кодировке UTF-8.

Текстовые файлы должны располагаться в одной папке и 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.

**!** Путь к текстовому корпусу должен содержать только стандартную латиницу.

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

Минимальный рекомендуемый размер пользовательского корпуса составляет 5000 строк.

Если пользовательский корпус имеет менее 5000 примеров, то будет выведено предупреждение о том, что обучение на таком корпусе будет малоэффективным.

#### Список зарезервированных (служебных) имен файлов:

Условные обозначения:

[*\$src*] – префикс входного языка

[*\$tgt*] – префикс выходного языка

aligned-grow-diag-final.realigned	corpus.bpe.[ <i>\$src</i> ]
corpus.bpe.[ <i>\$src</i> ].sub	corpus.bpe.[ <i>\$tgt</i> ]
corpus.bpe.[ <i>\$tgt</i> ].sub	corpus.[ <i>\$src</i> ]
corpus.[ <i>\$src</i> ].o	corpus.full.bpe.[ <i>\$src</i> ]
corpus.full.bpe.[ <i>\$tgt</i> ]	corpus.full.bpe.with.basic.[ <i>\$src</i> ]
corpus.full.bpe.with.basic.[ <i>\$tgt</i> ]	corpus.full.bpe.[ <i>\$src</i> ].dupl
corpus.full.bpe.[ <i>\$tgt</i> ].dupl	corpus.[ <i>\$tgt</i> ]
corpus.[ <i>\$tgt</i> ].o	corpus.tok.def.clean.[ <i>\$src</i> ]
corpus.tok.def.clean.[ <i>\$tgt</i> ]	corpus.tok.def.[ <i>\$src</i> ]
corpus.tok.def.[ <i>\$tgt</i> ]	corpus.tok.[ <i>\$src</i> ]
corpus.tok.[ <i>\$src</i> ].sub	corpus.tok.[ <i>\$tgt</i> ]
corpus.tok.[ <i>\$tgt</i> ].sub	aligned-grow-diag-final.realigned.dupl
general.[ <i>\$src</i> ].part	general.[ <i>\$tgt</i> ].part
general.tok.def.[ <i>\$src</i> ]	general.tok.def.[ <i>\$tgt</i> ]
general.tok.def.clean.[ <i>\$src</i> ]	general.tok.def.clean.[ <i>\$tgt</i> ]

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 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

**!** У пользователя, который запускает процесс настройки, должны быть права на запись в папку с текстовым корпусом.

## 5.2. Подготовка базовой модели

В качестве исходных данных для настройки модели используется архив базовой модели. NMT-модели поставляются в виде архивов с расширением .zip.

Скопируйте архив модели, которую необходимо обучить, в любую папку.

❗ У пользователя, который запускает процесс настройки, должны быть права на запись в папку с архивом модели.

✳ Путь к архиву с базовой моделью необходимо будет указать в файле **settings.json**. Подробнее – см. раздел [Запуск процесса настройки и задание параметров](#).

❗ Все указываемые пути к корпусам и моделям должны содержать только стандартную латиницу.

## 6. Настройка модели

### 6.1. Запуск процесса настройки и задание параметров

Процесс настройки модели запускается с помощью файла **run\_training.bat** из каталога установленного приложения (по умолчанию C:\Program Files (x86)\PROMT Neural Translation Server - Training Addon 23).

**!** Запуск *run\_training.bat* должен быть осуществлен с правами администратора: запустите сначала cmd.exe под администратором, а потом запустите из него bat-файл.

Перед запуском **run\_training.bat** отредактируйте файл **settings.json** (по умолчанию C:\Program Files (x86)\PROMT Neural Translation Server - Training Addon 23\utils\settings.json), задав параметры настройки модели через соответствующие переменные.

**Состав файла settings.json:**

<b>Source_Language</b>	Префикс входного языка.
<b>Target_Language</b>	Префикс выходного языка.
<b>Corpus</b>	Путь к папке с пользовательским корпусом. Путь должен содержать только стандартную латиницу.
<b>Validation_Percent</b>	Суммарный объем корпуса для валидации и тестового корпуса в процентах от исходного корпуса. При этом исходный обучающий корпус уменьшается на соответствующее количество строк. Рекомендуется задавать данный параметр в пределах от 1 до 10%. Суммарное максимальное число строк, выделяемое для двух этих корпусов, не может превышать 6000.
<b>Baseline_Model</b>	Путь к архиву с базовой моделью. Путь должен содержать только стандартную латиницу.
<b>Result_Model</b>	Путь к архиву с результирующей (настроенной) моделью. Проверьте, что папка, указанная для сохранения модели, существует.
<b>Model_Workspace</b>	Зарезервированный размер памяти (МБ), отведенный под процесс настройки модели. Если в качестве значения указано "auto", этот размер вычисляется перед запуском настройки модели, исходя из доступной памяти GPU устройства и из размера базовой модели.



	<p><b>Model_Workspace</b> не включает в себя память, отводимую под загрузку модели и работу marian.</p> <p>Рекомендуется подбирать данный параметр максимально большим, оставляя при этом свободными ~ 2.5 Гб на GPU устройстве.</p> <p>Если выделенный объем памяти слишком большой, процесс обучения будет остановлен с сообщением об ошибке.</p>
<b>Model_Valid_Freq</b>	<p>Частота валидаций настраиваемой модели в процессе настройки. Указывается число обновлений (updates), после которого необходимо запустить валидацию. Если в качестве значения указано "auto", значение выбирается автоматически (за весь процесс настройки будет выполнено 3 валидации).</p> <p>Порядок величин зависит от объема обучающего корпуса, выбранного значения параметра <b>Model_Workspace</b> и желаемого числа валидаций на весь процесс обучения.</p> <p>Рекомендуется подбирать данный параметр так, чтобы за весь процесс обучения выполнились 3-4 валидации. Большее число валидаций приводит к неоправданному увеличению времени обучения модели.</p>
<b>Model_GPU_Devices</b>	<p>Используемая видеокарта. Идентификатор GPU, используемого при обучении.</p> <p>Все GPU устройства, подключенные к компьютеру, получают числовые идентификаторы от 0 до N-1, где N – число GPU устройств на машине. Если указанный идентификатор выходит за границы диапазона, процесс обучения будет остановлен с сообщением об ошибке.</p>
<b>Training_Set_Min_Size</b>	<p>Минимальный размер, до которого дублируется пользовательский корпус. По умолчанию: 1 000 000 строк.</p>
<b>Python_path</b>	<p>Путь до python.exe.</p> <p>По умолчанию <i>C:\Program Files (x86)\PROMT Neural Translation Server - Training Addon 23\python3.7\python.exe</i></p>
<b>CT2</b>	<p>Автоматическая конвертация тренируемой модели marian в CT2. Если конвертация в CT2 необходима, укажите значение "1". Если для данного параметра указано значение "0", конвертация производится не будет.</p> <p>По умолчанию конвертация включена.</p>

Таблица 6.1. Параметры файла settings.json

✳ Если предполагается работать на системе с несколькими GPU, для выбора определенного устройства через параметр *Model\_GPU\_Devices* вызовите утилиту *promt-gpuinfo.exe*, которая находится в папке *Utils* установленного аддона. Запуск утилиты следует осуществлять из командной строки.

! Для обучения моделей следует выбирать только одно устройство GPU.

! Все указываемые пути к корпусам и моделям должны содержать только стандартную латиницу.

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

### Пример файла `settings.json` для настройки англо-русской модели:

```
{
  "Source_Language": "en",
  "Target_Language": "ru",
  "Corpus": "C:\\Users\\Admin\\Corpus",
  "Baseline_Model": "C:\\Users\\Admin\\er_20.5_07.20.zip ",
  "Model_Workspace": "auto",
  "Model_CPU_Threads": "0",
  "Model_GPU_Devices": "0",
  "Validation_Percent": "10",
  "Model_Valid_Freq": "auto",
  "Result_Model": "C:\\Users\\Admin\\er_20.5_07.20.zip -training.zip",
  "Training_Set_Min_Size": "1000000",
  "Python_path": "C:\\Program Files (x86)\\PROMT Neural Translation Server - Training
Addon 23\\python3.7\\python.exe",
  "CT2": "1"
}
```

## 6.2. Этапы обучения модели

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

### 6.2.1. Препроцессинг

Препроцессинг состоит из следующих шагов:

#### 1. Создание исходного пользовательского 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**. 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 *prompt.nts.cleancorpus.exe* в процессе работы отфильтровывает и выводит в отдельный лог *corpus.tok.def.clean.badstring.txt* строки, которые не могут быть использованы для обучения модели. Лог сохраняется в папке с пользовательским корпусом.

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 **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-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 **prompt.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 **prompt.nts.tokenize**. На выходе получаются обработанные файлы основного сета, синтетического сета и сета *dev* с дополнительной строкой *.bre* в названии.

## 7. 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`.

## 8. Подсчет количества необходимых итераций для обучения

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

### 6.2.2. 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 **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.

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`;

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 `settings.json`. Конвертация выполняется утилитой `prompt.nts.ct2converter.exe`.

В ходе выполнения настройки создаются файлы с логами:

- `train.log` – вывод информации о ходе обучения;
- `valid.log` – информация о валидациях.

✳ Обученную NMT-модель можно подгрузить в PNTS средствами, предусмотренными в установленной версии PNTS. Подробнее об установке NMT-моделей см. руководство администратора PNTS.

## 7. Вычисление финального значения BLEU

Подсчет финального значения BLEU производится с помощью утилиты **promt.nts.BSUtils.exe**, запускаемой командным файлом **run\_bleuscore.bat**.

Утилита вычисляет значение BLEU, сравнивая эталонный перевод с результатом машинного перевода исходного текста, выполненного *PNTS* с заданным профилем перевода.

 Запуск файла *run\_bleuscore.bat* должен осуществляться из командной строки

Для оценки результатов настройки модели выполните следующее:

1. Вычислите значение BLEU для базовой модели.

Для этого запустите файл **run\_bleuscore.bat** с нужными параметрами и получите значение BLEU, сравнивая эталонный перевод (*test.\$tgt* из *test-сета*) с результатом машинного перевода исходного текста (*test.\$src* из *test-сета*), выполненного с профилем перевода, в котором подключена базовая модель.

2. Вычислите значение BLEU для настроенной модели.

Для этого запустите файл **run\_bleuscore.bat** с нужными параметрами и получите значение BLEU, сравнивая эталонный перевод (*test.\$tgt* из *test-сета*) с результатом машинного перевода исходного текста (*test.\$src* из *test-сета*), выполненного с профилем перевода, в котором подключена настроенная модель.

3. Сравните полученные результаты BLEU.

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.


Значение BLEU выдается как результат работы утилиты в консоль.

### 7.1. Параметры для запуска утилиты

Для запуска утилиты в командном файле **run\_bleuscore.bat** необходимо задать ряд параметров:

- **PTS\_URL** – строка с URL PNTS (например, SET PTS\_URL=http://localhost/as);
- **PTS\_LINUX** – использование Windows- или Linux-версии сервера перевода. Значения:
  - 0** – используется Windows-версия сервера перевода PNTS;
  - 23** – используется PNTS Linux версии 23 (например, PNTS23.1 - в таком случае параметр будет выглядеть так: PTS\_LINUX=23);
  - 22** – используется PNTS Linux версии 22 (например, PNTS22.1 - в таком случае параметр будет выглядеть так: PTS\_LINUX=22).
- **PTS\_USER** – пользователь PNTS (для случая Forms аутентификации);

- **PTS\_USER\_PASSWORD** - пароль (для случая Forms аутентификации);
- **SRC** – префикс исходного языка;
- **TGT** – префикс языка перевода;
- **TOPIC** – профиль перевода, с которым будет производится перевод в PNTS;
- **SRC\_FILE** – путь к исходному текстовому файлу test.\$src из test-сета (кодировка UTF-8).
- **REF\_FILE** – путь к текстовому файлу с эталонным переводом test.\$tgt из test-сета (кодировка UTF-8).

 Во время подсчета BLEU для базовой и обученной NMT-моделей не рекомендуется изменять конфигурацию компьютера и/или настройки PNTS.

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

Пример файла run\_bleuscore.bat для вычисления значения BLEU для англо-русской модели, подключенной в профиле "test":

```
@echo off
chcp 65001
SET PTS_URL=http://localhost/as
SET PTS_USER=
SET PTS_USER_PASSWORD=
SET SRC=en
SET TGT=ru
SET TOPIC=test
SET REF_FILE=c:\temp\corpus\test.ru
SET SRC_FILE=c:\temp\corpus\test.en
SET PTS_LINUX=0

@echo on
"%~dp0utils\promt.nts.bsutil.exe" --ref="%REF_FILE%" --sf="%SRC_FILE%" --
url=%PTS_URL% "--topic=%TOPIC%" --src=%SRC% --tgt=%TGT% --user="%PTS_USER%" --
password="%PTS_USER_PASSWORD%" --linux="%PTS_LINUX%" --
tokenizer="%~dp0utils\promt.nts.tokenize.exe"
```

## 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 `mapian` (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` (пример: перевод с `!!!gaw!!!`); 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** – 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	no
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
Шведский	sv
Эстонский	et
Японский	ja