DeepPavlov

Release 0.17.3

Neural Networks and Deep Learning lab, MIPT

Apr 28, 2022

CONTENTS

CHAPTER ONE

GET STARTED WITH DEEPPAVLOV LIBRARY

1.1 Installation

[1]:

```
!pip install deeppavlov
```

```
Requirement already satisfied: deeppavlov in /home/docs/checkouts/readthedocs.org/user_
↪builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (0.17.3)
Requirement already satisfied: nltk==3.4.5 in /home/docs/checkouts/readthedocs.org/user_
↪builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov) (3.4.
↪5)
Requirement already satisfied: pymorphy2==0.8 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(0.8)
Requirement already satisfied: pymorphy2-dicts-ru in /home/docs/checkouts/readthedocs.
↪org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from␣
↪deeppavlov) (2.4.417127.4579844)
Requirement already satisfied: aio-pika==6.4.1 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(6.4.1)
Requirement already satisfied: sacremoses==0.0.35 in /home/docs/checkouts/readthedocs.
↪org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from␣
↪deeppavlov) (0.0.35)
Requirement already satisfied: overrides==2.7.0 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(2.7.0)
Requirement already satisfied: pytz==2019.1 in /home/docs/checkouts/readthedocs.org/user_
↪builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov) (2019.
↪1)
Requirement already satisfied: rusenttokenize==0.0.5 in /home/docs/checkouts/readthedocs.
↪org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from␣
↪deeppavlov) (0.0.5)
Requirement already satisfied: ruamel.yaml==0.15.100 in /home/docs/checkouts/readthedocs.
↪org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from␣
↪deeppavlov) (0.15.100)
Requirement already satisfied: pandas==0.25.3 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(0.25.3)
Requirement already satisfied: click==7.1.2 in /home/docs/checkouts/readthedocs.org/user_
↪builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov) (7.1.
↪2)
Requirement already satisfied: fastapi==0.47.1 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(0.47.1)
```

(continues on next page)

(continued from previous page)

```
Requirement already satisfied: pytelegrambotapi==3.6.7 in /home/docs/checkouts/
↪readthedocs.org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages␣
↪(from deeppavlov) (3.6.7)
Requirement already satisfied: scikit-learn==0.21.2 in /home/docs/checkouts/readthedocs.
↪org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from␣
↪deeppavlov) (0.21.2)
Requirement already satisfied: prometheus-client==0.7.1 in /home/docs/checkouts/
↪readthedocs.org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages␣
↪(from deeppavlov) (0.7.1)
Requirement already satisfied: filelock==3.0.12 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(3.0.12)
Requirement already satisfied: pydantic==1.3 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(1.3)
Requirement already satisfied: uvicorn==0.11.7 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(0.11.7)
Requirement already satisfied: requests==2.22.0 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(2.22.0)
Requirement already satisfied: h5py==2.10.0 in /home/docs/checkouts/readthedocs.org/user_
↪builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov) (2.10.
↪0)
Requirement already satisfied: tqdm==4.62.0 in /home/docs/checkouts/readthedocs.org/user_
↪builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov) (4.62.
↪0)
Requirement already satisfied: numpy==1.18.0 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(1.18.0)
Requirement already satisfied: Cython==0.29.14 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(0.29.14)
Requirement already satisfied: scipy==1.4.1 in /home/docs/checkouts/readthedocs.org/user_
↪builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov) (1.4.
↪1)
Requirement already satisfied: uvloop==0.14.0 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(0.14.0)
Requirement already satisfied: pyopenssl==22.0.0 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from deeppavlov)␣
↪(22.0.0)
Requirement already satisfied: yarl in /home/docs/checkouts/readthedocs.org/user_builds/
↪deeppavlov-test/envs/latest/lib/python3.7/site-packages (from aio-pika==6.4.1->
↪deeppavlov) (1.7.2)
Requirement already satisfied: aiormq<4,>=3.2.0 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from aio-pika==6.
↪4.1->deeppavlov) (3.3.1)
Requirement already satisfied: starlette<=0.12.9,>=0.12.9 in /home/docs/checkouts/
↪readthedocs.org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages␣
↪(from fastapi==0.47.1->deeppavlov) (0.12.9)
Requirement already satisfied: six in /home/docs/checkouts/readthedocs.org/user_builds/
↪deeppavlov-test/envs/latest/lib/python3.7/site-packages (from h5py==2.10.0->
↪deeppavlov) (1.16.0)
```

(continues on next page)

(continued from previous page)

```
Requirement already satisfied: python-dateutil>=2.6.1 in /home/docs/checkouts/
↪readthedocs.org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages␣
↪(from pandas==0.25.3->deeppavlov) (2.8.2)
Requirement already satisfied: docopt>=0.6 in /home/docs/checkouts/readthedocs.org/user_
↪builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from pymorphy2==0.8->
↪deeppavlov) (0.6.2)
Requirement already satisfied: dawg-python>=0.7 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from pymorphy2==0.
↪8->deeppavlov) (0.7.2)
Requirement already satisfied: pymorphy2-dicts<3.0,>=2.4 in /home/docs/checkouts/
↪readthedocs.org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages␣
↪(from pymorphy2==0.8->deeppavlov) (2.4.393442.3710985)
Requirement already satisfied: cryptography>=35.0 in /home/docs/checkouts/readthedocs.
↪org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from␣
↪pyopenssl==22.0.0->deeppavlov) (37.0.1)
Requirement already satisfied: idna<2.9,>=2.5 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from requests==2.
↪22.0->deeppavlov) (2.8)
Requirement already satisfied: certifi>=2017.4.17 in /home/docs/checkouts/readthedocs.
↪org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from␣
↪requests==2.22.0->deeppavlov) (2021.10.8)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /home/docs/
↪checkouts/readthedocs.org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-
↪packages (from requests==2.22.0->deeppavlov) (1.25.11)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /home/docs/checkouts/readthedocs.
↪org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from␣
↪requests==2.22.0->deeppavlov) (3.0.4)
Requirement already satisfied: joblib in /home/docs/checkouts/readthedocs.org/user_
↪builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from sacremoses==0.0.
↪35->deeppavlov) (1.1.0)
Requirement already satisfied: httptools==0.1.* in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from uvicorn==0.
↪11.7->deeppavlov) (0.1.2)
Requirement already satisfied: websockets==8.* in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from uvicorn==0.
↪11.7->deeppavlov) (8.1)
Requirement already satisfied: h11<0.10,>=0.8 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from uvicorn==0.
↪11.7->deeppavlov) (0.9.0)
Requirement already satisfied: pamqp==2.3.0 in /home/docs/checkouts/readthedocs.org/user_
↪builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from aiormq<4,>=3.2.0->
↪aio-pika==6.4.1->deeppavlov) (2.3.0)
Requirement already satisfied: cffi>=1.12 in /home/docs/checkouts/readthedocs.org/user_
↪builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from cryptography>=35.
↪0->pyopenssl==22.0.0->deeppavlov) (1.15.0)
Requirement already satisfied: multidict>=4.0 in /home/docs/checkouts/readthedocs.org/
↪user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from yarl->aio-
↪pika==6.4.1->deeppavlov) (6.0.2)
Requirement already satisfied: typing-extensions>=3.7.4 in /home/docs/checkouts/
↪readthedocs.org/user_builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages␣
↪(from yarl->aio-pika==6.4.1->deeppavlov) (4.2.0)
Requirement already satisfied: pycparser in /home/docs/checkouts/readthedocs.org/user_
↪builds/deeppavlov-test/envs/latest/lib/python3.7/site-packages (from cffi>=1.12->
↪cryptography>=35.0->pyopenssl==22.0.0->deeppavlov) (2.21)
```

(continues on next page)

(continued from previous page)

To be continued

[ ]:

```
```

CHAPTER

TWO

WHAT IS A CONFIG FILE?

Here we will provide an elaborate description of the main concepts of training and building models from a config file, their structure and variables.

[ ]:

```
```

CHAPTER

THREE

NAMED ENTITY RECOGNITION (NER)

3.1 Table of contents

3.2 1. Introduction to the task

Named Entity Recognition (NER) is a task of assigning a tag (from a predefined set of tags) to each token in a given sequence. In other words, NER-task consists of identifying named entities in the text and classifying them into types (e.g. person name, organization, location etc).

BIO encoding schema is usually used in NER task. It uses 3 tags: B for the beginning of the entity, I for the inside of the entity, and O for non-entity tokens. The second part of the tag stands for the entity type.

Here is an example of a tagged sequence:

Elon	Musk	founded	Tesla	in	2003	.
B-PER	I-PER	O	B-ORG	O	B-DATE	O

Here we can see three extracted named entities: *Elon Musk* (which is a person's name), *Tesla* (which is a name of an organization) and *2003* (which is a date). To see more examples try out our Demo.

The list of possible types of NER entities may vary depending on your dataset domain. The list of tags used in DeepPavlov's models can be found in the *table*.

3.3 2. Get started with the model

First make sure you have the DeepPavlov Library installed. More info about the first installation

```
[1]: !pip install --q deeppavlov
```

Then make sure that all the required packages for the model are installed.

```
[ ]: !python -m deeppavlov install ner_ontonotes_bert_torch
```

`ner_ontonotes_bert_torch` here is the name of the model's *config_file*. What is a Config File?

Configuration file defines the model and describes its hyperparameters. To use another model, change the name of the *config_file* here and further. The full list of NER models with their config names can be found in the *table*.

3.4 3. Use the model for prediction

3.4.1 3.1 Predict using Python

After *installing* the model, build it from the config and predict.

```
[ ]: from deeppavlov import configs, build_model

     ner_model = build_model(configs.ner.ner_ontonotes_bert_torch, download=True)
```

Input: List[sentences]

Output: List[tokenized sentences, corresponding NER-tags]

```
[ ]: ner_model(['Bob Ross lived in Florida', 'Elon Musk founded Tesla'])
```

```
[[['Bob', 'Ross', 'lived', 'in', 'Florida'],
  ['Elon', 'Musk', 'founded', 'Tesla']],
 [['B-PERSON', 'I-PERSON', 'O', 'O', 'B-GPE'],
  ['B-PERSON', 'I-PERSON', 'O', 'B-ORG']]]
```

3.4.2 3.2 Predict using CLI

You can also get predictions in an interactive mode through CLI.

```
[ ]: ! python -m deeppavlov interact ner_ontonotes_bert_torch -d
```

`-d` is an optional download key (alternative to `download=True` in Python code). The key `-d` is used to download the pre-trained model along with embeddings and all other files needed to run the model.

Or make predictions for samples from *stdin*.

[ ]:

```
! python -m deeppavlov predict ner_ontonotes_bert_torch -f <file-name>
```

3.5 4. Evaluate

There are two metrics that are used to evaluate a NER model in DeepPavlov:

`ner_f1` is measured on the entity-level (actual text spans should match exactly)

`ner_token_f1` is measured on a token level (correct tokens from not fully extracted entities will still be counted as TPs)

3.5.1 4.1 Evaluate from Python

[ ]:

```
from deeppavlov import evaluate_model

model = evaluate_model(configs.ner.ner_ontonotes_bert_torch, download=True)
```

3.5.2 4.1 Evaluate from CLI

[ ]:

```
! python -m deeppavlov evaluate ner_ontonotes_bert_torch
```

3.6 5. Train the model on your data

3.6.1 5.1 Train your model from Python

Provide your data path

To train the model on your data, you need to change the path to the training data in the *config_file*.

Parse the *config_file* and change the path to your data from Python.

[ ]:

```
from deeppavlov import configs, train_model
from deeppavlov.core.commands.utils import parse_config

model_config = parse_config(configs.ner.ner_ontonotes_bert_torch)

#  dataset that the model was trained on
print(model_config['dataset_reader']['data_path'])
```

```
~/.deeppavlov/downloads/ontonotes/
```

Provide a *data_path* to your own dataset.

[7]:

```
# download and unzip a new example dataset
!wget http://files.deeppavlov.ai/deeppavlov_data/conll2003_v2.tar.gz
!tar -xzvf "conll2003_v2.tar.gz"
```

[6]:
```
# provide a path to the train file
model_config["dataset_reader"]["data_path"] = "contents/train.txt"
```

Train dataset format

To train the model, you need to have a txt-file with a dataset in the following format:

```
EU B-ORG
rejects O
the O
call O
of O
Germany B-LOC
to O
boycott O
lamb O
from O
Great B-LOC
Britain I-LOC
. O

China B-LOC
says O
time O
right O
for O
Taiwan B-LOC
talks O
. O
```

The source text is **tokenized** and **tagged**. For each token, there is a tag with BIO markup. Tags are separated from tokens with **whitespaces**. Sentences are separated with **empty lines**.

Train the model using new config

[ ]:
```
ner_model = train_model(model_config)
```

Use your model for prediction.

[ ]:
```
ner_model(['Bob Ross lived in Florida', 'Elon Musk founded Tesla'])
```

```
[[['Bob', 'Ross', 'lived', 'in', 'Florida'],
  ['Elon', 'Musk', 'founded', 'Tesla']],
 [['B-PERSON', 'I-PERSON', 'O', 'O', 'B-GPE'],
  ['B-PERSON', 'I-PERSON', 'O', 'B-ORG']]]
```

3.6.2 5.2 Train your model from CLI

```
[ ]: ! python -m deeppavlov train ner_ontonotes_bert_torch
```

3.7 6. Models list

The table presents a list of all of the NER-models available in DeepPavlov Library.

Config name	Dataset	Language	Model Size	F1 score
ner_lcquad_bert_ent_and_type	LC-QuAD	En	?	?
ner_conll2003_bert	CoNLL-2003	En	1.3 GB	90.7
ner_ontonotes_bert	Ontonotes	En	1.3 GB	87.9
ner_ontonotes_bert_probas	Ontonotes	En	?	?
ner_ontonotes_bert_mult	Ontonotes	Multi	2.0 GB	87.2
ner_bert_ent_and_type_rus	LC-QuAD Rus	Ru	?	?
ner_rus_bert	Collection3	Ru	2.0 GB	97.7
ner_rus_bert_probas	Collection3	Ru	?	?
ner_collection3_bert	Collection3	Ru	1.0 GB	97.8
ner_rus_convers_distilrubert_2L	Collection3	Ru	?	?
ner_rus_convers_distilrubert_6L	Collection3	Ru	?	?

3.8 7. NER-tags list

The table presents a list of all of the NER entity tags used in DeepPavlov's NER-models.

PERSON	People including fictional
NORP	Nationalities or religious or political groups
FACILITY	Buildings, airports, highways, bridges, etc.
ORGANIZATION	Companies, agencies, institutions, etc.
GPE	Countries, cities, states
LOCATION	Non-GPE locations, mountain ranges, bodies of water
PRODUCT	Vehicles, weapons, foods, etc. (Not services)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK OF ART	Titles of books, songs, etc.
LAW	Named documents made into laws
LANGUAGE	Any named language
DATE	Absolute or relative dates or periods
TIME	Times smaller than a day
PERCENT	Percentage (including "%")
MONEY	Monetary values, including unit
QUANTITY	Measurements, as of weight or distance
ORDINAL	"first", "second"
CARDINAL	Numerals that do not fall under another type

CHAPTER

FOUR

CLASSIFICATION

4.1 Table of contents

4.2 1. Introduction to the task

This section describes a family of BERT-based models that solve a variety of different classification tasks.

Insults detection is a binary classification task of identying wether a given sequence is an insult of another participant of communication.

Sentiment analysis is a task of classifying the polarity of the the given sequence. The number of classes may vary depending on the data: positive/negative binary classification, multiclass classification with a neutral class added or with a number of different emotions.

The models trained for the **paraphrase detection** task identify whether two sentences expressed with different words convey the same meaning.

4.3 2. Get started with the model

First make sure you have the DeepPavlov Library installed. More info about the first installation

```
[ ]: !pip install --q deeppavlov
```

Then make sure that all the required packages for the model are installed.

```
[ ]: !python -m deeppavlov install insults_kaggle_bert_torch
```

`insults_kaggle_bert_torch` here is the name of the model's *config_file*. What is a Config File?

Configuration file defines the model and describes its hyperparameters. To use another model, change the name of the *config_file* here and further. The full list of NER models with their config names can be found in the *table*.

4.4 3. Use the model for prediction

4.4.1 3.1 Predict using Python

After *installing* the model, build it from the config and predict.

```
[ ]: from deeppavlov import configs, build_model

     model = build_model(configs.classifiers.insults_kaggle_bert_torch, download=True)
```

Input format: List[sentences]

Output format: List[labels]

```
[4]: model(['You are kind of stupid', 'You are a wonderful person!'])
```

```
[4]: ['Insult', 'Not Insult']
```

4.4.2 3.2 Predict using CLI

You can also get predictions in an interactive mode through CLI.

```
[ ]: ! python deeppavlov interact insults_kaggle_bert_torch -d
```

`-d` is an optional download key (alternative to `download=True` in Python code). The key `-d` is used to download the pre-trained model along with embeddings and all other files needed to run the model.

Or make predictions for samples from *stdin*.

```
[ ]: ! python deeppavlov predict insults_kaggle_bert_torch -f <file-name>
```

4.5 4. Evaluation

4.5.1 4.1 Evaluate from Python

[ ]:

```
from deeppavlov import evaluate_model

model = evaluate_model(configs.classifiers.insults_kaggle_bert_torch, download=True)
```

4.5.2 4.1 Evaluate from CLI

[ ]:

```
! python -m deeppavlov evaluate insults_kaggle_bert_torch -d
```

4.6 5. Train the model on your data

4.6.1 5.1 Train your model from Python

Provide your data path

To train the model on your data, you need to change the path to the training data in the *config_file*.

Parse the *config_file* and change the path to your data from Python.

[6]:

```
from deeppavlov import configs, train_model
from deeppavlov.core.commands.utils import parse_config

model_config = parse_config(configs.classifiers.insults_kaggle_bert_torch)

#  dataset that the model was trained on
print(model_config['dataset_reader']['data_path'])
```

```
~/.deeppavlov/downloads/insults_data
```

Provide a *data_path* to your own dataset. You can also change any of the hyperparameters of the model.

[ ]:

```
# download and unzip a new example dataset
!wget http://files.deeppavlov.ai/datasets/insults_data.tar.gz
!tar -xzvf "insults_data.tar.gz"
```

[ ]:

```
# provide a path to the directory with your train, valid and test files
model_config["dataset_reader"]["data_path"] = "./contents/"
```

Train dataset format

Train the model using new config

```
[ ]: model = train_model(model_config)
```

Use your model for prediction.

```
[5]: model(['You are kind of stupid', 'You are a wonderful person!'])
```

```
[5]: ['Insult', 'Not Insult']
```

4.6.2 5.2 Train your model from CLI

To train the model on your data, create a copy of a config file and change the *data_path* variable in it. After that, train the model using your new *config_file*. You can also change any of the hyperparameters of the model.

```
[ ]: ! python -m deeppavlov train model_config.json
```

4.7 6. Models list

The table presents a list of all of the classification models available in DeepPavlov Library.

Config name	Task	Dataset	Language	Model Size	Score
insults_kaggle_bert	Insult Detection	Insults	En	1.1 GB	ROC-AUC: 0.877
paraphraser_bert	Paraphrase Detection	?	En	?	?
paraphraser_convers_distilrubert_2L	Paraphrase Detection	?	En	?	?
paraphraser_convers_distilrubert_6L	Paraphrase Detection	?	En	?	?
paraphraser_rubert	Paraphrase Detection	?	Ru	?	?
sentiment_sst_conv_bert	Sentiment Analysis	SST	En	?	?
rusentiment_bert	Sentiment Analysis	RuSentiment	Ru	?	?
rusentiment_convers_bert	Sentiment Analysis	RuSentiment	Ru	?	?
rusentiment_convers_distilrubert_2L	Sentiment Analysis	RuSentiment	Ru	?	?
rusentiment_convers_distilrubert_6L	Sentiment Analysis	RuSentiment	Ru	?	?
sentiment_twitter	Sentiment Analysis	Twitter Data	Ru?	?	?

CHAPTER

FIVE

CONTEXT QUESTION ANSWERING

5.1 Table of contents

5.2 1. Introduction to the task

Context Question Answering is a task of finding a fragment with an answer to a question in a given segment of context.

Context:

```
In meteorology, precipitation is any product of the condensation
of atmospheric water vapor that falls under gravity. The main forms
of precipitation include drizzle, rain, sleet, snow, graupel and hail...
Precipitation forms as smaller droplets coalesce via collision with
other rain drops or ice crystals within a cloud. Short, intense periods
of rain in scattered locations are called “showers”.
```

Question:

```
Where do water droplets collide with ice crystals to form precipitation?
```

Answer:

```
within a cloud
```

Datasets that follow this task format:

- Stanford Question Answering Dataset (SQuAD) (EN)
- SberQuAD (RU)

5.3 2. Get started with the model

First make sure you have the DeepPavlov Library installed. More info about the first installation

```
[ ]: !pip install --q deeppavlov
```

Then make sure that all the required packages for the model are installed.

```
[ ]: !python -m deeppavlov install squad_torch_bert
```

`squad_torch_bert` here is the name of the model's *config_file*. What is a Config File?

Configuration file defines the model and describes its hyperparameters. To use another model, change the name of the *config_file* here and further. The full list of the models with their config names can be found in the *table*.

5.4 3. Use the model for prediction

5.4.1 3.1 Predict using Python

After *installing* the model, build it from the config and predict.

```
[ ]: from deeppavlov import configs, build_model

     model = build_model(configs.squad.squad_torch_bert, download=True)
```

Input: List[context, question]

Output: List[answer, start_character, logit]

```
[ ]: model(['DeepPavlov is a library for NLP and dialog systems.'], ['What is DeepPavlov?'])
```

```
[['a library for NLP and dialog systems'], [14], [200928.390625]]
```

5.4.2 3.2 Predict using CLI

You can also get predictions in an interactive mode through CLI.

[ ]:

```
!python -m deeppavlov interact squad_torch_bert -d
```

`-d` is an optional download key (alternative to `download=True` in Python code). The key `-d` is used to download the pre-trained model along with embeddings and all other files needed to run the model.

Or make predictions for samples from *stdin*.

[ ]:

```
!python -m deeppavlov predict squad_torch_bert -f <file-name>
```

5.5 4. Train the model on your data

5.5.1 4.1 Train your model from Python

Provide your data path

To train the model on your data, you need to change the path to the training data in the *config_file*.

Parse the *config_file* and change the path to your data from Python.

[ ]:

```
from deeppavlov import configs, train_model
from deeppavlov.core.commands.utils import parse_config

model_config = parse_config(configs.squad.squad_torch_bert)

#  dataset that the model was trained on
print(model_config['dataset_reader']['data_path'])
```

```
~/.deeppavlov/downloads/squad/
```

Provide a *data_path* to your own dataset.

[2]:

```
# download and unzip a new example dataset
!wget http://files.deeppavlov.ai/datasets/squad-v1.1.tar.gz
!tar -xzvf "squad-v1.1.tar.gz"
```

Note that if you want to provide your own dataset, it should have the same format as the SQuAD dataset downloaded in this cell.

[ ]:

```
# provide a path to the train file
model_config["dataset_reader"]["data_path"] = "/contents/train-v1.1.json"
```

SQuAD dataset info

There are *two* versions of the SQuAD dataset available for training at the moment:

- SQuAD 1.1 contains 107,785 question-answer pairs on 536 articles. Dataset size: `33.52 MiB`.
- SQuAD 2.0 combines all of the questions from SQuAD 1.1 with over 50,000 un-answerable questions written adversarially by crowdworkers. Dataset size: `44.34 MiB`.

Train the model using new config

[ ]:

```
model = train_model(model_config)
```

Use your model for prediction.

[ ]:

```
model(['DeepPavlov is a library for NLP and dialog systems.'], ['What is DeepPavlov?'])
```

```
[['a library for NLP and dialog systems'], [14], [200928.390625]]
```

5.5.2 4.2 Train your model from CLI

[ ]:

```
!python -m deeppavlov train squad_torch_bert
```

5.6 5. Evaluate

5.6.1 5.1 Evaluate from Python

[ ]:

```
from deeppavlov import evaluate_model

model = evaluate_model(configs.squad.squad_torch_bert, download=True)
```

5.6.2 5.1 Evaluate from CLI

[ ]:

```
! python -m deeppavlov evaluate squad_torch_bert -d
```

5.7 6. Models list

The table presents a list of all of the Context Question Answering models available in DeepPavlov Library.

Config name	Dataset	Language	Model Size	F1 score
squad_torch_bert	SQuAD	En	? GB	?
multi_squad_noans	SQuAD	Multi	? GB	?
multi_squad_retr_noans	SQuAD	Multi	? GB	?
squad_bert_multilingual_freezed_emb	SQuAD	Multi	? GB	?
multi_squad_ru_retr_noans	SQuAD	?	? GB	?
multi_squad_ru_retr_noans_rubert	SQuAD	?	? GB	?
squad_ru_bert	SQuAD	Ru	? GB	?
squad_ru_convers_distilrubert_2L	SQuAD	Ru	? GB	?
squad_ru_convers_distilrubert_6L	SQuAD	Ru	? GB	?
squad_ru_rubert	SberQuAD	Ru	? GB	?
squad_torch_bert	SberQuAD	Ru	? GB	?

[ ]: