Klasyfikacja całych słów

Zagdnieniem jakie poruszymy w tym rodziale jest rozpoznawanie całych słów zapisanych pismem odręcznym.

Rozpoczynam od załadowania niezbędnych bibliotek. Będziemy używać biblioteki “Mxnet” oraz “GluonNLP” do trenowania sieci neuronowej, która posłuży do rozpoznawania pisma odręcznego. Dane testowe pochodzą ze zbioru odręcznie zapisanych formularzy iam.

Do przygotowania zbioru danych użyjemy biblioteki pochodzącej ze zbioru AWS o nazwie OCR. Dzięki tej bibliotece możemy wczytać zbiór i przygotować go do przetwarzania. Ważne jest aby w pliku credentials.json określić dane dostępowe do api serwisu iam.

[1]:
```python
### Import  bibliotek

import random

random.seed(123)

import matplotlib.pyplot as plt
import matplotlib.patches as patches
import mxnet as mx
import gluonnlp as nlp
import numpy as np
import cv2 as cv
from skimage import transform as skimage_tf, exposure
from tqdm import tqdm

from ocr.utils.expand_bounding_box import expand_bounding_box
from ocr.utils.sclite_helper import ScliteHelper
from ocr.utils.iam_dataset import IAMDataset, resize_image, crop_image,␣
 ↪crop_handwriting_page
from ocr.utils.word_to_line import sort_bbs_line_by_line, crop_line_images
from ocr.paragraph_segmentation_dcnn import SegmentationNetwork,␣
 ↪paragraph_segmentation_transform
from ocr.utils.encoder_decoder import Denoiser, ALPHABET, encode_char,␣
 ↪decode_char, EOS, BOS

from ocr.utils.denoiser_utils import SequenceGenerator

from ocr.utils.beam_search import ctcBeamSearch

from ocr.word_and_line_segmentation import SSD as WordSegmentationNet,␣
 ↪predict_bounding_boxes
from ocr.handwriting_line_recognition import Network as␣
 ↪HandwritingRecognitionNet, handwriting_recognition_transform
from ocr.handwriting_line_recognition import decode as decoder_handwriting,␣
 ↪alphabet_encoding
```

```
[07:20:20] ../src/c_api/../operator/custom/custom-inl.h:57: New registration is
overriding existing custom operator _smoothing_with_dim
```

[2]:
```python
ctx = mx.gpu(0)
```

Przygotowanie zbioru danych.

Badanie przeprowadzimy na dwóch przygotowanych przeze mnie formularz. Jeden formularz został zapisany w języku angielskim, a drugi w języku polskim.

[3]:

```
import json

files = [
    'A00-000',
    'A00-002',
    'A00-003',
    'A00-001'
]

words = json.load(open('my-dataset/words.json',))

test_ds = []

for x in files:
    img = cv.imread(f'my-dataset/{x}.png')[:, :, 0]
    img = np.array(img)
    text = np.array(words[x])
    test_ds.append([img, text])
```

[4]:

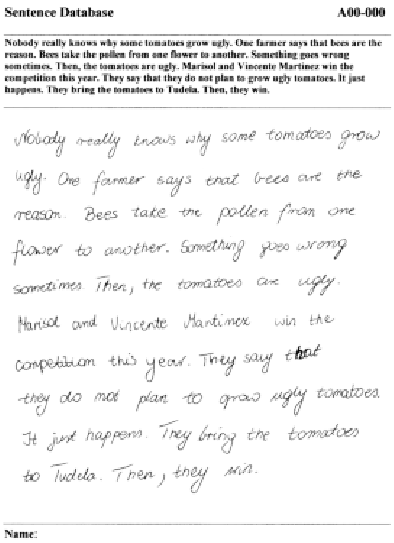
```
random.seed(1)
```

[5]:

```
figs_to_plot = 4
images = []

n = 0
for i in range(0, figs_to_plot):
    image, _ = test_ds[i]
    images.append(image)
```

[6]:

```
fig, axs = plt.subplots(int(len(images)/2), 2, figsize=(15, 10 * len(images)/
 ↪2))
for i, image in enumerate(images):
    y, x = int(i/2), int(i%2)
    axs[y, x].imshow(image, cmap='Greys_r')
    axs[y, x].axis('off')
```

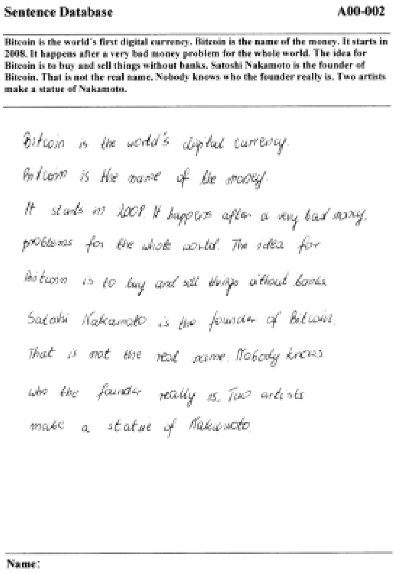


Sentence Database A00-000

Nobody really knows why some tomatoes grow ugly. One farmer says that bees are the reason. Bees take the pollen from one flower to another. Something goes wrong sometimes. Then, the tomatoes are ugly. Marisol and Vincente Martinez win the competition this year. They say that they do not plan to grow ugly tomatoes. It just happens. They bring the tomatoes to Tudela. Then, they win.

Nobody really knows why some tomatoes grow
ugly. One farmer says that bees are the
reason. Bees take the pollen from one
flower to another. Something goes wrong
sometimes. Then, the tomatoes are ugly.
Marisol and Vincente Martinez win the
competition this year. They say that
they do not plan to grow ugly tomatoes.
It just happens. They bring the tomatoes
to Tudela. Then, they win.

Name:

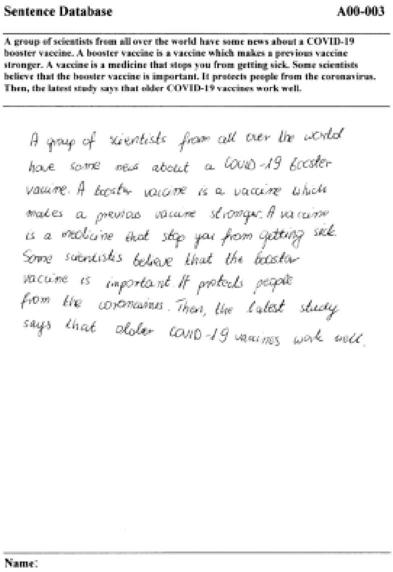


Sentence Database A00-002

Bitcoin is the world´s first digital currency. Bitcoin is the name of the money. It starts in 2008. It happens after a very bad money problem for the whole world. The idea for Bitcoin is to buy and sell things without banks. Satoshi Nakamoto is the founder of Bitcoin. That is not the real name. Nobody knows who the founder really is. Two artists make a statue of Nakamoto.

Bitcoin is the world's digital currency.
Bitcoin is the name of the money.
It starts in 2008. It happens after a very bad money
problems for the whole world. The idea for
Bitcoin is to buy and sell things without banks.
Satoshi Nakamoto is the founder of Bitcoin.
That is not the real name. Nobody knows
who the founder really is. Two artists
make a statue of Nakamoto.

Name:

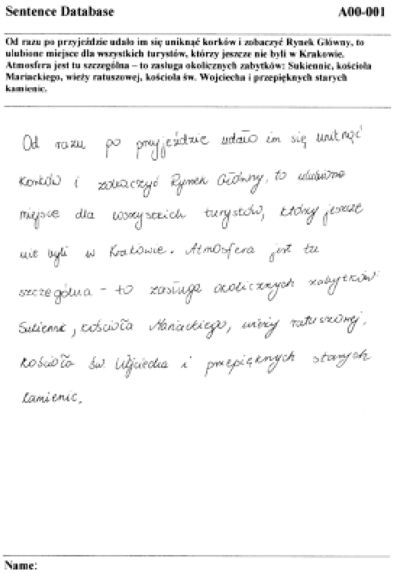


Sentence Database A00-003

A group of scientists from all over the world have some news about a COVID-19 booster vaccine. A booster vaccine is a vaccine which makes a previous vaccine stronger. A vaccine is a medicine that stops you from getting sick. Some scientists believe that the booster vaccine is important. It protects people from the coronavirus. Then, the latest study says that older COVID-19 vaccines work well.

A group of scientists from all over the world
have some news about a COVID-19 booster
vaccine. A booster vaccine is a vaccine which
makes a previous vaccine stronger. A vaccine
is a medicine that stop you from getting sick.
Some scientists believe that the booster
vaccine is important. It protects people
from the coronavirus. Then, the latest study
says that older COVID-19 vaccines work well.

Name:



Sentence Database A00-001

Od razu po przyjeździe udało im się uniknąć korków i zobaczyć Rynek Główny, to ulubione miejsce dla wszystkich turystów, którzy jeszcze nie byli w Krakowie. Atmosfera jest tu szczególna – to zasługa okolicznych zabytków: Sukiennic, kościoła Mariackiego, wieży ratuszowej, kościoła św. Wojciecha i przepięknych starych kamienic.

Od razu po przyjeździe udało im się uniknąć
korków i zobaczyć Rynek Główny, to ulubione
miejsce dla wszystkich turystów, którzy jeszcze
nie byli w Krakowie. Atmosfera jest tu
szczególna - to zasługa okolicznych zabytków:
Sukiennic, kościoła Mariackiego, wieży ratuszowej,
kościoła św. Wojciecha i przepięknych starych
kamienic.

Name:

Segmentacja akapitów

Mając obraz formularza w zbiorze danych IAM, należy przewidzieć obszar w którym występuje tekst pisany odręcznie. Model został przeszkolony przy użyciu algorytmu omówionego w notatniku segmentacja akapitów.

[7]:

```
paragraph_segmentation_net = SegmentationNetwork(ctx=ctx)
paragraph_segmentation_net.cnn.load_parameters("models/paragraph_segmentation2.
 ↪params", ctx=ctx)
```

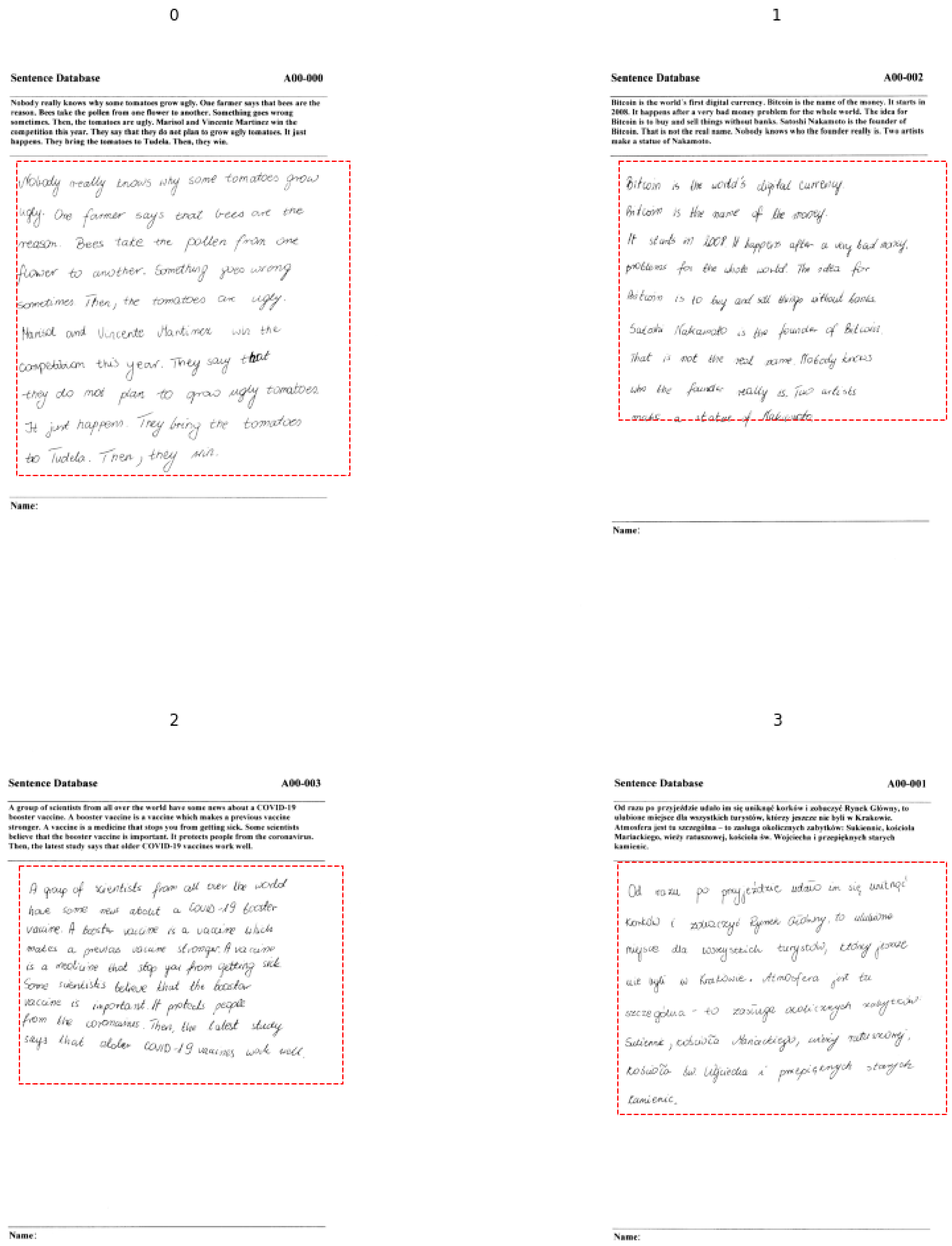
[8]:

```
paragraph_segmentation_net.hybridize()
```

[9]:

```
form_size = (1120, 800)

predicted_bbs = []

fig, axs = plt.subplots(int(len(images)/2), 2, figsize=(15, 9 * len(images)/2))
for i, image in enumerate(images):
    s_y, s_x = int(i/2), int(i%2)
    resized_image = paragraph_segmentation_transform(image, form_size)
    bb_predicted = paragraph_segmentation_net(resized_image.as_in_context(ctx))
    bb_predicted = bb_predicted[0].asnumpy()
    bb_predicted = expand_bounding_box(bb_predicted, expand_bb_scale_x=0.03,
                                        expand_bb_scale_y=0.03)
    predicted_bbs.append(bb_predicted)

    axs[s_y, s_x].imshow(image, cmap='Greys_r')
    axs[s_y, s_x].set_title("{}".format(i))

    (x, y, w, h) = bb_predicted
    image_h, image_w = image.shape[-2:]
    (x, y, w, h) = (x * image_w, y * image_h, w * image_w, h * image_h)
    rect = patches.Rectangle((x, y), w, h, fill=False, color="r", ls="--")
    axs[s_y, s_x].add_patch(rect)
    axs[s_y, s_x].axis('off')
```

```
[07:20:40] ../src/operator/nn/./cudnn/./cudnn_algoreg-inl.h:97: Running␣
 ↪performance tests
to find the best convolution algorithm, this can take a while… (set the␣
 ↪environment
variable MXNET_CUDNN_AUTOTUNE_DEFAULT to 0 to disable)
```



Sentence Database A00-000

Nobody really knows why some tomatoes grow ugly. One farmer says that bees are the reason. Bees take the pollen from one flower to another. Something goes wrong sometimes. Then, the tomatoes are ugly. Marisol and Vincente Martinez win the competition this year. They say that they do not plan to grow ugly tomatoes. It just happens. They bring the tomatoes to Tudela. Then, they win.

Nobody really knows why some tomatoes grow
ugly. One farmer says that bees are the
reason. Bees take the pollen from one
flower to another. Something goes wrong
sometimes. Then, the tomatoes are ugly.
Marisol and Vincente Martinez win the
competition this year. They say that
they do not plan to grow ugly tomatoes
It just happens. They bring the tomatoes
to Tudela. Then, they win.

Name:

Sentence Database A00-002

Bitcoin is the world's first digital currency. Bitcoin is the name of the money. It starts in 2008. It happens after a very bad money problem for the whole world. The idea for Bitcoin is to buy and sell things without banks. Satoshi Nakamoto is the founder of Bitcoin. That is not the real name. Nobody knows who the founder really is. Two artists make a statue of Nakamoto.

Bitcoin is the world's digital currency.
Bitcoin is the name of the money.
It starts in 2008. It happens after a very bad money
problems for the whole world. The idea for
Bitcoin is to buy and sell things without banks.
Satoshi Nakamoto is the founder of Bitcoin.
That is not the real name. Nobody knows
who the founder really is. Two artists
make a statue of Nakamoto.

Name:

Sentence Database A00-003

A group of scientists from all over the world have some news about a COVID-19 booster vaccine. A booster vaccine is a vaccine which makes a previous vaccine stronger. A vaccine is a medicine that stops you from getting sick. Some scientists believe that the booster vaccine is important. It protects people from the coronavirus. Then, the latest study says that older COVID-19 vaccines work well.

A group of scientists from all over the world
have some news about a COVID-19 booster
vaccine. A booster vaccine is a vaccine which
makes a previous vaccine stronger. A vaccine
is a medicine that stop you from getting sick
Some scientists believe that the booster
vaccine is important. It protects people
from the coronavirus. Then, the latest study
says that older COVID-19 vaccines work well.

Name:

Sentence Database A00-001

Od razu po przyjeździe udało im się uniknąć korków i zobaczyć Rynek Główny, to ulubione miejsce dla wszystkich turystów, którzy jeszcze nie byli w Krakowie. Atmosfera jest tu szczególna – to zasługa okolicznych zabytków: Sukiennic, kościoła Mariackiego, wieży ratuszowej, kościoła św. Wojciecha i przepięknych starych kamienic.

Od razu po przyjeździe udało im się uniknąć
korków i zobaczyć Rynek Główny, to ulubione
miejsce dla wszystkich turystów, którzy jeszcze
nie byli w Krakowie. Atmosfera jest tu
szczególna - to zasługa okolicznych zabytków:
Sukiennic, kościoła Mariackiego, wieży ratuszowej,
kościoła św. Wojciecha i przepięknych starych
kamienic.

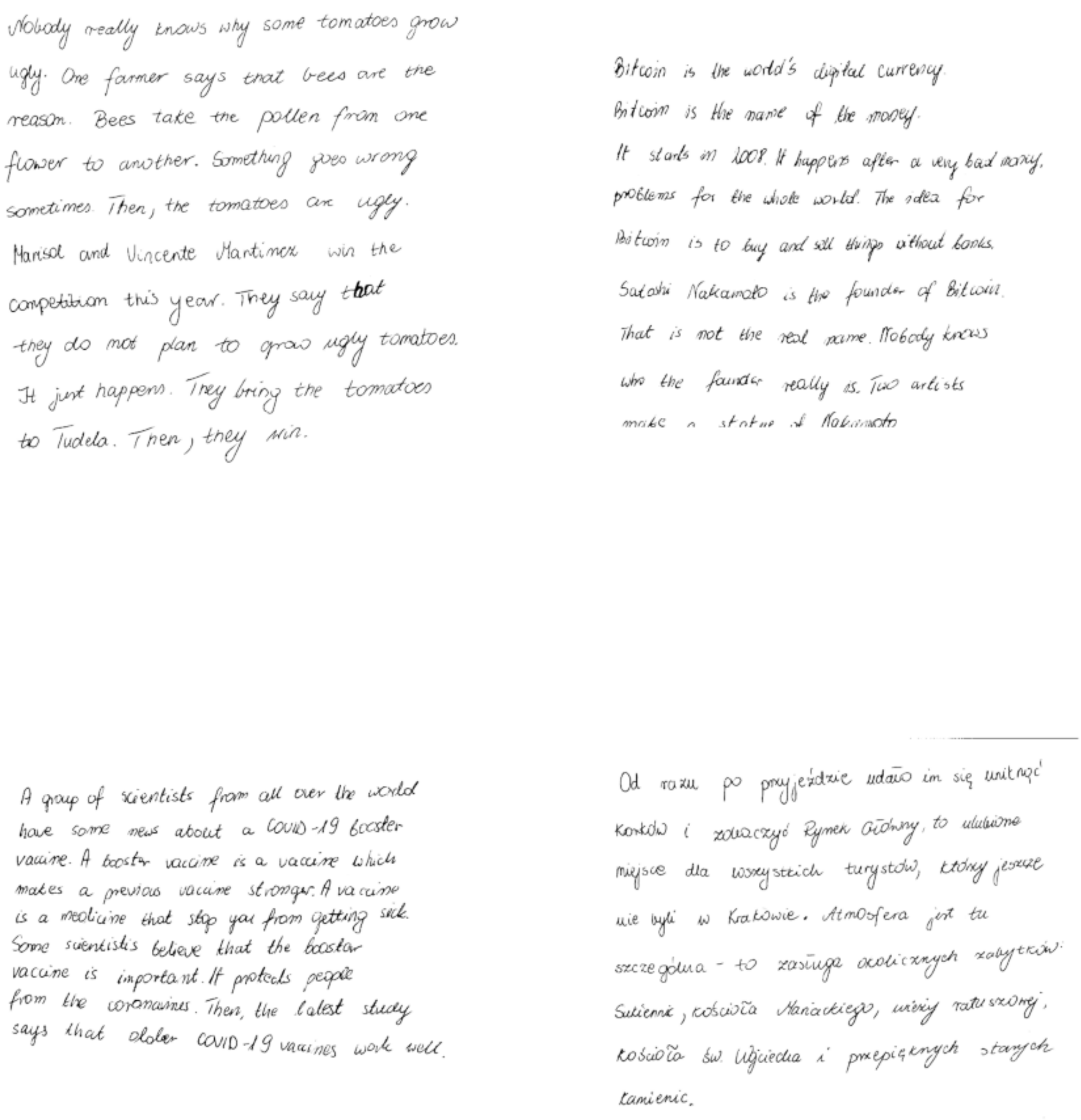
Name:

Przygotowanie zdjęć

Wycinamy z obrazu tylko ramkę z ręcznie pisanym tekstem i dostarczamy ją dalej do algorytmu.

[10]:

```
segmented_paragraph_size = (700, 700)
fig, axs = plt.subplots(int(len(images)/2), 2, figsize=(15, 9 * len(images)/2))

paragraph_segmented_images = []

for i, image in enumerate(images):
    s_y, s_x = int(i/2), int(i%2)

    bb = predicted_bbs[i]
    image = crop_handwriting_page(image, bb,␣
↪image_size=segmented_paragraph_size)
    paragraph_segmented_images.append(image)

    axs[s_y, s_x].imshow(image, cmap='Greys_r')
    axs[s_y, s_x].axis('off')
```



Nobody really knows why some tomatoes grow
ugly. One farmer says that bees are the
reason. Bees take the pollen from one
flower to another. Something goes wrong
sometimes. Then, the tomatoes are ugly.
Marisol and Vincente Martinez win the
competition this year. They say that
they do not plan to grow ugly tomatoes.
It just happens. They bring the tomatoes
to Tudela. Then, they win.

Bitcoin is the world's digital currency.
Bitcoin is the name of the money.
It starts in 2008. It happens after a very bad money,
problems for the whole world. The idea for
Bitcoin is to buy and sell things without banks.
Satoshi Nakamoto is the founder of Bitcoin.
That is not the real name. Nobody knows
who the founder really is. Two artists
make a statue of Nakamoto

A group of scientists from all over the world
have some news about a COVID-19 booster
vaccine. A booster vaccine is a vaccine which
makes a previous vaccine stronger. A vaccine
is a medicine that stop you from getting sick.
Some scientists believe that the booster
vaccine is important. It protects people
from the coronavirus. Then, the latest study
says that older COVID-19 vaccines work well.

Od razu po przyjeździe udało im się unitnąć
Kraków i zobaczyć Rynek Główny, to ulubione
miejsce dla wszystkich turystów, którzy jeszcze
nie byli w Krakowie. Atmosfera jest tu
szczególna - to zasługa okolicznych zabytków:
Sukiennic, kościoła Mariackiego, wieży ratuszowej,
kościoła św. Wojciecha i przepięknych starych
kamienic.

Segmentacja lini tesktu i słów

Mając formularz zawierający wyłącznie tekst pisany odręcznie, należy przewidzieć ramkę dla każdego słowa. Sposób działania modelu został przedstawiony w notatniku segmentacja tekstu i słów.

[11]:

```
word_segmentation_net = WordSegmentationNet(2, ctx=ctx)
word_segmentation_net.load_parameters("models/word_segmentation2.params")
word_segmentation_net.hybridize()
```

[12]:

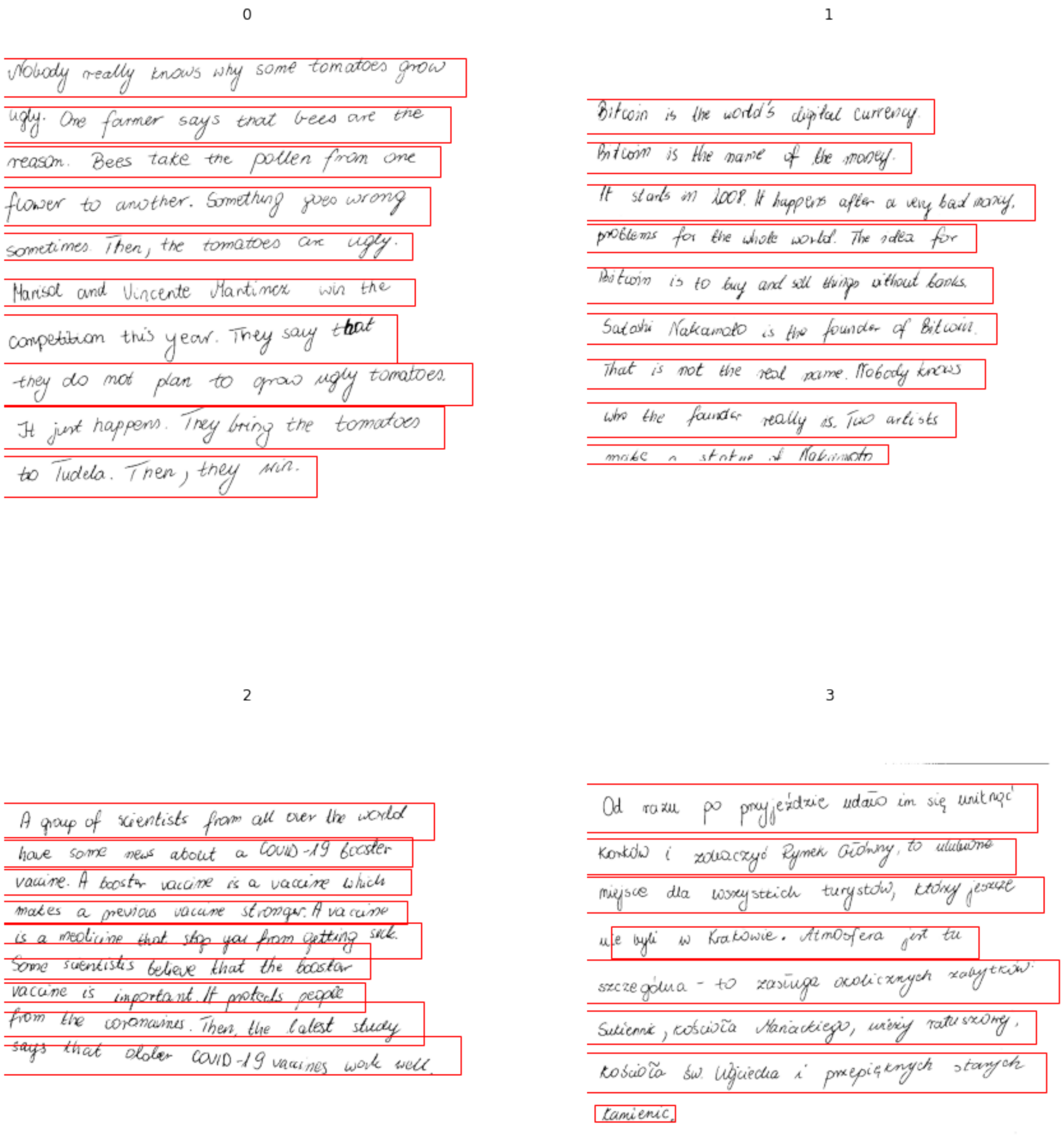
```
min_c = 0.1
overlap_thres = 0.1
topk = 600

fig, axs = plt.subplots(int(len(paragraph_segmented_images)/2), 2,
                        figsize=(15, 5 * int(len(paragraph_segmented_images)/
 ↪2)))
predicted_words_bbs_array = []

for i, paragraph_segmented_image in enumerate(paragraph_segmented_images):
    s_y, s_x = int(i/2), int(i%2)

    predicted_bb = predict_bounding_boxes(
        word_segmentation_net, paragraph_segmented_image, min_c,␣
 ↪overlap_thres, topk, ctx)

    predicted_words_bbs_array.append(predicted_bb)

    axs[s_y, s_x].imshow(paragraph_segmented_image, cmap='Greys_r')
    for j in range(predicted_bb.shape[0]):
        (x, y, w, h) = predicted_bb[j]
        image_h, image_w = paragraph_segmented_image.shape[-2:]
        (x, y, w, h) = (x * image_w, y * image_h, w * image_w, h * image_h)
        rect = patches.Rectangle((x, y), w, h, fill=False, color="r")
        axs[s_y, s_x].add_patch(rect)
        axs[s_y, s_x].axis('off')
```

Dopasowanie pojedyńczych słów do lini

Następnie musimy z tekstu wydobyć całe lini posiadające tekst odreczny.

[13]:

```
line_images_array = []
fig, axs = plt.subplots(int(len(paragraph_segmented_images)/2), 2,
                        figsize=(15, 9 * int(len(paragraph_segmented_images)/
 ↪2)))

for i, paragraph_segmented_image in enumerate(paragraph_segmented_images):
    s_y, s_x = int(i/2), int(i%2)
    axs[s_y, s_x].imshow(paragraph_segmented_image, cmap='Greys_r')
    axs[s_y, s_x].axis('off')
    axs[s_y, s_x].set_title("{}".format(i))

    predicted_bbs = predicted_words_bbs_array[i]
    line_bbs = sort_bbs_line_by_line(predicted_bbs, y_overlap=0.4)
    line_images = crop_line_images(paragraph_segmented_image, line_bbs)
    line_images_array.append(line_images)

    for line_bb in line_bbs:
        (x, y, w, h) = line_bb
        image_h, image_w = paragraph_segmented_image.shape[-2:]
        (x, y, w, h) = (x * image_w, y * image_h, w * image_w, h * image_h)

        rect = patches.Rectangle((x, y), w, h, fill=False, color="r")
        axs[s_y, s_x].add_patch(rect)
```


0
Nobody really knows why some tomatoes grow
ugly. One farmer says that bees are the
reason. Bees take the pollen from one
flower to another. Something goes wrong
sometimes. Then, the tomatoes are ugly.
Marisol and Vincente Martinez win the
competition this year. They say that
they do not plan to grow ugly tomatoes.
It just happens. They bring the tomatoes
to Tudela. Then, they win.
1
Bitcoin is the world's digital currency.
Bitcoin is the name of the money.
It starts in 2008. It happens after a very bad money
problems for the whole world. The idea for
Bitcoin is to buy and sell things without banks.
Satoshi Nakamoto is the founder of Bitcoin.
That is not the real name. Nobody knows
who the founder really is. Two artists
make a statue of Nakamoto
2
A group of scientists from all over the world
have some news about a COVID-19 booster
vaccine. A booster vaccine is a vaccine which
makes a previous vaccine stronger. A vaccine
is a medicine that stops you from getting sick.
Some scientists believe that the booster
vaccine is important. It protects people
from the coronavirus. Then, the latest study
says that older COVID-19 vaccines work well.
3
Od razu po przyjeździe udało im się uniknąć
Korków i zobaczyć Rynek Główny, to ulubione
miejsce dla wszystkich turystów, którzy jeszcze
nie byli w Krakowie. Atmosfera jest tu
szczególna - to zasługa okolicznych zabytków:
Sukiennic, kościoła Mariackiego, wieży ratuszowej,
kościoła św. Wojciecha i przepięknych starych
kamienic.

Rozpoznawanie pisma

Biorąc pod uwagę każdy wiersz tekstu, przewidujemy ciąg tekstu pisanego odręcznie. Działanie tej sieci został przedstawione w notatniku rozpoznawanie pisma.

[14]:
```
handwriting_line_recognition_net = HandwritingRecognitionNet(rnn_hidden_states=512,
                                                              rnn_layers=2, ctx=ctx, max_seq_len=160)
handwriting_line_recognition_net.load_parameters("models/handwriting_line8.params", ctx=ctx)
handwriting_line_recognition_net.hybridize()
```

[15]:
```
line_image_size = (60, 800)
character_probs = []
for line_images in line_images_array:
    form_character_prob = []
    for i, line_image in enumerate(line_images):
        line_image = handwriting_recognition_transform(line_image,␣
 ↪line_image_size)
        line_character_prob = handwriting_line_recognition_net(line_image.
 ↪as_in_context(ctx))
        form_character_prob.append(line_character_prob)
    character_probs.append(form_character_prob)
```

Prawdopodobieństwo znaków w tekście

[16]:
```
def get_arg_max(prob):
    '''
    The greedy algorithm convert the output of the handwriting recognition␣
 ↪network
    into strings.
    '''
    arg_max = prob.topk(axis=2).asnumpy()
    return decoder_handwriting(arg_max)[0]
```

[17]:
```
def get_beam_search(prob, width=5):
    possibilities = ctcBeamSearch(prob.softmax()[0].asnumpy(),␣
 ↪alphabet_encoding, None, width)
    return possibilities[0]
```

Odszumianie tekstu wyjściowego

Używamy denoisera seq2seq, aby przetłumaczyć zaszumione wejście na lepsze jakościowo wyjście.

[18]:
```
FEATURE_LEN = 150
denoiser = Denoiser(alphabet_size=len(ALPHABET), max_src_length=FEATURE_LEN,␣
 ↪max_tgt_length=FEATURE_LEN, num_heads=16, embed_size=256, num_layers=2)
denoiser.load_parameters('models/denoiser2.params', ctx=ctx)
```

[19]:
```
denoiser.hybridize(static_alloc=True)
```

[20]:
```
ctx_nlp = mx.gpu(0)
language_model, vocab = nlp.model.big_rnn_lm_2048_512(dataset_name='gbw',␣
 ↪pretrained=True, ctx=ctx_nlp)
moses_tokenizer = nlp.data.SacreMosesTokenizer()
moses_detokenizer = nlp.data.SacreMosesDetokenizer()
```

[21]:
```
beam_sampler = nlp.model.BeamSearchSampler(beam_size=20,
                                           decoder=denoiser.decode_logprob,
                                           eos_id=EOS,
                                           scorer=nlp.model.BeamSearchScorer(),
                                           max_length=150)
```

[22]:
```
generator = SequenceGenerator(beam_sampler, language_model, vocab, ctx_nlp,␣
 ↪moses_tokenizer, moses_detokenizer)
```

[23]:
```
def get_denoised(prob, ctc_bs=False):
    if ctc_bs: # Using ctc beam search before denoising yields only limited␣
 ↪improvements a is very slow
```

```
        text = get_beam_search(prob)
    else:
        text = get_arg_max(prob)
    src_seq, src_valid_length = encode_char(text)
    src_seq = mx.nd.array([src_seq], ctx=ctx)
    src_valid_length = mx.nd.array(src_valid_length, ctx=ctx)
    encoder_outputs, _ = denoiser.encode(src_seq,␣
↪valid_length=src_valid_length)
    states = denoiser.decoder.init_state_from_encoder(encoder_outputs,
                                                              ␣
↪encoder_valid_length=src_valid_length)
    inputs = mx.nd.full(shape=(1,), ctx=src_seq.context, dtype=np.float32,␣
↪val=BOS)
    output = generator.generate_sequences(inputs, states, text)
    return output.strip()
```

[24]:
```
sentence = "This sentnce has an eror"
src_seq, src_valid_length = encode_char(sentence)
src_seq = mx.nd.array([src_seq], ctx=ctx)
src_valid_length = mx.nd.array(src_valid_length, ctx=ctx)
encoder_outputs, _ = denoiser.encode(src_seq, valid_length=src_valid_length)
states = denoiser.decoder.init_state_from_encoder(encoder_outputs,
                                                          ␣
↪encoder_valid_length=src_valid_length)
inputs = mx.nd.full(shape=(1,), ctx=src_seq.context, dtype=np.float32, val=BOS)
print(sentence)
print("Choice")
print(generator.generate_sequences(inputs, states, sentence))
```

```
This sentnce has an eror
Choice
T h i s   s e n t e n c e
```

Wyniki jakościowe

- [AM] Arg Max CTC Decoding
- [BS] Beam Search CTC Decoding
- [D] Adding Text Denoiser

[25]:
```
for i, form_character_probs in enumerate(character_probs):
    fig, axs = plt.subplots(len(form_character_probs) + 1,
                            figsize=(10, int(1 + 2.3 *␣
↪len(form_character_probs))))
    for j, line_character_probs in enumerate(form_character_probs):
        decoded_line_am = get_arg_max(line_character_probs)
        print("[AM]",decoded_line_am)
        decoded_line_bs = get_beam_search(line_character_probs)
        decoded_line_denoiser = get_denoised(line_character_probs,␣
↪ctc_bs=False)
        print("[D ]",decoded_line_denoiser)

        line_image = line_images_array[i][j]
        axs[j].imshow(line_image.squeeze(), cmap='Greys_r')
        axs[j].set_title("\n\n\n\n[AM]: {}\n[BS]: {}\n[D ]: {}\n".
↪format(decoded_line_am, decoded_line_bs, decoded_line_denoiser),␣
↪fontdict={"horizontalalignment":"left", "family":"monospace"}, x=0)
        axs[j].axis('off')
    axs[-1].imshow(np.zeros(shape=line_image_size), cmap='Greys_r')
    axs[-1].axis('off')
```

```
[AM] vlobody ceally knows why some tomatoes grow
[D ] s o m e b o d y   r e a l l y   k n o w s   w
[AM] ngly. Ohe fammer says that bees ane the
[D ] u g l y .   T h e   f a r m e r   s a y s
[AM] reasam. Bees take the pollen from come
[D ] r e a s o n .   B e e s   t a k e   t h
[AM] flomser to another. Something gues wrong
[D ] f l o w e r   t o   a n o t h e r .   S o
[AM] sometimes. Thery the tomatoes are ugly.
[D ] s o m e t i m e s .   T h e r e b y   t h
[AM] Marieal and Vincente Mantimere win the
[D ] M a r i e l   a n d   V i n c e n t e
[AM] competitian this yeorr. They sany that
[D ] c o m p e t i t i o n   t h i s   y e a
[AM] they do not plan to grous mgly tomatoes.
[D ] t h e y   d o   n o t   p l a n   t o   g
[AM] It just happens. They bring the tomatues
[D ] I t   j u s t   h a p p e n s .   T h e y
[AM] to Tudela. Ther; they sin-
[D ] t o   J u d e a .   T h e n
[AM] Bitcain is the workd's digital currewcs.
[D ] B r i t a i n   i s   t h e   w o r l d '
[AM] Britcin is the marre of the mosery.
[D ] B r i t a i n   i s   t h e   m e r e
[AM] It starks in 1L00). H happerts aflen on very bard marcy.
[D ] I t   s h a r k s   i n   1 0 0 ) .   H e   s u p p o r t
[AM] probrems for tne whole workd. The idea for
[D ] p r o b l e m s   f o r   t h e   w h o l e
[AM] Britain is to buy and well things without Banks.
[D ] Britain is to buy and well things without Banks.
[AM] Satolti Maxcamatd is the foundior of Bitcain.
[D ] S a t a l l i   M e x i c a m a t e d   i n   t
[AM] That is not the seal name. Maborly knows
[D ] T h a t   i s   n o t   t h e   r e a l
[AM] wtho the foundar really as Two arlistes
[D ] w i t h o u t   t h e   f o u n d e r   r
[AM] mate a stnturd it Malismats
[D ] m a d e   a   s t a n d a r d
[AM] A group of sccentists from all over the world
[D ] A   g r o u p   o f   s c i e n t i s t s   f r
[AM] have some news about a coun9-49 bccster
[D ] h a v e   s o m e   n e w s   a b o u t
[AM] vacaine. A coostor vacaine is a vacime which
[D ] v a c i n e .   A   c o a s t   o r   v a c i
```

```
[AM] mates a previous vacime stronger. A vaccinse
[D ] m a t e s   a   p r e v i o u s   v a c i m e
[AM] is a melicine, that stop you from getting side.
[D ] i s   m e d i c i n e ,   t h a t   s t o p   y o
[AM] Somic scensistis" Soviere" that" the Soostert
[D ] S o m i c   a s c e n s i s t   i s   " S o v e
[AM] raccine, is imporeant. It protects people
[D ] p r a c t i c e ,   i s   i m p o t e n t .
[AM] from the corioiaines. Then, the "atest stuaey
[D ] f r o m   t h e   c o n t r a d i c t i n e s .
[AM] says that ololen covio. 1 9vacines worke were.
[D ] s a y s   t h a t   w h o l e n   c o n v i c t
[AM] Od natn po pruyjesiduic udaro in sie unilme.
[D ] h a d   n o t h i n g   u p o n   y o u r   p
[AM] Konkohs i rdkacryd aymen Grduny, to mlulcieme
[D ] m o n k o u s   i n   o r d i n a r y   d a y
[AM] miejoce da worysttich turystdws, ktdry jesuse
[D ] w h i c h   s i n c e   w a s   a   v e r y   s
[AM] ie uyin is trationie. Aemoofera just th
[D ] l i k e   b u y i n g   h i s   t r a d i
[AM] secregama - to rasinge orolicanych rabytrcin"
[D ] s e c r e g a m e   -   t o   r i s i n g   t h
[AM] Suxiennic, robcidia Mariactiegs, wieing ratuseoney.
[D ] S u s a n n i c ,   s o l i c i t i c i a l   M a r c
[AM] Rosccia du. Uigciecka is preepierengch stawyoh
[D ] R e b e c c i a   d u .   M i c n i c k a   i s
[AM] Camienic.
[D ] C a m e r i c .
```

```
[AM]: vlobody ceally knows why some tomatoes grow
[BS]: vlobody ceally knows why some tomatoes grow
[D ]: s o m e b o d y   r e a l l y   k n o w s   w
```

Nobody really knows why some tomatoes grow

```
[AM]: ngly. Ohe fammer says that bees ane the
[BS]: ngly. Ohe farmer says that bees ane the
[D ]: u g l y .   T h e   f a r m e r   s a y s
```

ugly. One farmer says that bees are the

```
[AM]: reasam. Bees take the pollen from come
[BS]: reasam. Bees take the pollen from come
[D ]: r e a s o n .   B e e s   t a k e   t h
```

reason. Bees take the pollen from one

```
[AM]: flomser to another. Something gues wrong
[BS]: flomser to another. Something gues wrong
[D ]: f l o w e r   t o   a n o t h e r .   S o
```

flower to another. Something goes wrong

```
[AM]: sometimes. Thery the tomatoes are ugly.
[BS]: sometimes. Thery the tomatoes are ugly.
[D ]: s o m e t i m e s .   T h e r e b y   t h
```

sometimes. Then, the tomatoes are ugly.

```
[AM]: Marieal and Vincente Mantimere win the
[BS]: Marieal and Vincente Mantimere win the
[D ]: M a r i e l   a n d   V i n c e n t e
```

Marisol and Vincente Martinez win the

```
[AM]: competitian this yeorr. They sany that
[BS]: competitian this yeorr. They sany that
[D ]: c o m p e t i t i o n   t h i s   y e a
```

competition this year. They say that

```
[AM]: they do not plan to grous mgly tomatoes.
[BS]: they do not plan to grous mgly tomatoes.
[D ]: t h e y   d o   n o t   p l a n   t o   g
```

they do not plan to grow ugly tomatoes.

```
[AM]: It just happens. They bring the tomatues
[BS]: It just happens. They bring the tomatues
[D ]: I t   j u s t   h a p p e n s .   T h e y
```

It just happens. They bring the tomatoes

```
[AM]: to Tudela. Ther; they sin-
[BS]: to Tudela. Ther; they sin-
[D ]: t o   J u d e a .   T h e n
```

to Tudela. Then, they win.

[redacted]

```
[AM]: Bitcain is the workd's digital currewcs.
[BS]: Bitcain is the workd's digital currewcs.
[D ]: B r i t a i n   i s   t h e   w o r l d '
```

Bitcoin is the world's digital currency.

```
[AM]: Britcin is the marre of the mosery.
[BS]: Britcin is the marre of the mosey.
[D ]: B r i t a i n   i s   t h e   m e r e
```

Bitcoin is the name of the money.

```
[AM]: It starks in 1L00). H happerts aflen on very bard marcy.
[BS]: It starks in 100). H happerts aflen on very bard marcy.
[D ]: I t   s h a r k s   i n   1 0 0 ) .   H e   s u p p o r t
```

It starts in 2008. It happens after a very bad money.

```
[AM]: probrems for tne whole workd. The idea for
[BS]: probrems for tne whole workd. The idea for
[D ]: p r o b l e m s   f o r   t h e   w h o l e
```

problems for the whole world. The idea for

```
[AM]: Britain is to buy and well things without Banks.
[BS]: Britain is to buy and well things without Banks.
[D ]: Britain is to buy and well things without Banks.
```

Bitcoin is to buy and sell things without banks.

```
[AM]: Satolti Maxcamatd is the foundior of Bitcain.
[BS]: Satolti Maxcamatd is the foundior of Bitcain.
[D ]: S a t a l l i   M e x i c a m a t e d   i n   t
```

Satoshi Nakamoto is the founder of Bitcoin.

```
[AM]: That is not the seal name. Maborly knows
[BS]: That is not the seal name. Maborly knows
[D ]: T h a t   i s   n o t   t h e   r e a l
```

That is not the real name. Nobody knows

```
[AM]: wtho the foundar really as Two arlistes
[BS]: wtho the foundar really as Two arlistes
[D ]: w i t h o u t   t h e   f o u n d e r   r
```

who the founder really is. Two artists

```
[AM]: mate a stnturd it Malismats
[BS]: mate a stnturd it Malismats
[D ]: m a d e   a   s t a n d a r d
```

make a statue of Nakamoto

[redacted]

```
[AM]: A group of sccentists from all over the world
[BS]: A group of sccentists from all over the world
[D ]: A  g r o u p  o f  s c i e n t i s t s  f r
```

A group of scientists from all over the world

```
[AM]: have some news about a coun9-49 bccster
[BS]: have some news about a coun9-49 bccster
[D ]: h a v e  s o m e  n e w s  a b o u t
```

have some news about a COVID-19 booster

```
[AM]: vacaine. A coostor vacaine is a vacime which
[BS]: vacaine. A coostor vacaine is a vacime which
[D ]: v a c i n e .  A  c o a s t  o r  v a c i
```

vaccine. A booster vaccine is a vaccine which

```
[AM]: mates a previous vacime stronger. A vaccinse
[BS]: mates a previous vacime stronger. A vaccinse
[D ]: m a t e s  a  p r e v i o u s  v a c i m e
```

makes a previous vaccine stronger. A vaccine

```
[AM]: is a melicine, that stop you from getting side.
[BS]: is a melicine, that stop you from getting side.
[D ]: i s  m e d i c i n e ,  t h a t  s t o p  y o
```

is a medicine that stop you from getting sick.

```
[AM]: Somic scensistis" Soviere" that" the Soostert
[BS]: Somic sciencistis" Soviere" that" the Soostert
[D ]: S o m i c  a s c e n s i s t  i s  " S o v e
```

Some scientistis believe that the booster

```
[AM]: raccine, is imporeant. It protects people
[BS]: raccine, is imporeant. It protects people
[D ]: p r a c t i c e ,  i s  i m p o t e n t .
```

vaccine is important. It protects people

```
[AM]: from the corioiaines. Then, the "atest stuaey
[BS]: from the corioinaines. Then, the "atest stuaey
[D ]: f r o m  t h e  c o n t r a d i c t i n e s .
```

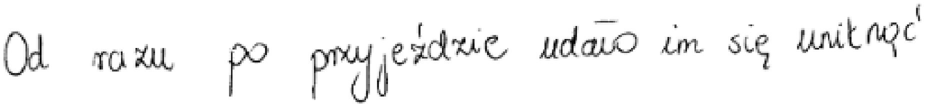
from the coronavirus. Then, the latest study

```
[AM]: says that ololen covio. 1 9vacines worke were.
[BS]: says that ololen covio. 1 9vacines worke were.
[D ]: s a y s  t h a t  w h o l e n  c o n v i c t
```

says that older COVID-19 vaccines work well.

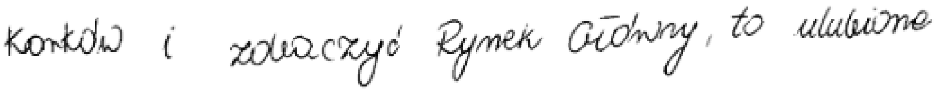
[redacted]

```
[AM]: Od natn po pruyjesiduic udaro in sie unilme.
[BS]: Od natn po pruyjesiduic udaro in sie unilme.
[D ]: had nothing upon your p
```



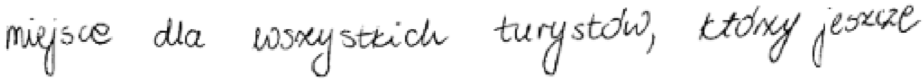
Od razu po pryyjeździe udało im się umknąć

```
[AM]: Konkohs i rdkacryd aymen Grduny, to mlulcieme
[BS]: Konkohs i rdkacryd aymen Grduny, to mlulcieme
[D ]: monkous in ordinary day
```



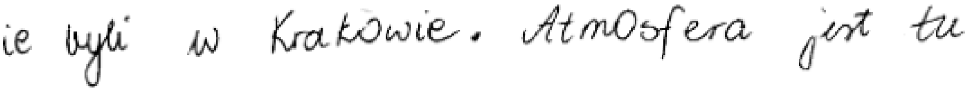
Konków i zobaczyć Rynek Główny, to ulubione

```
[AM]: miejoce da worysttich turystdws, ktdry jesuse
[BS]: miejoce da worysttich turystdws, ktdry jesuse
[D ]: which since was a very s
```



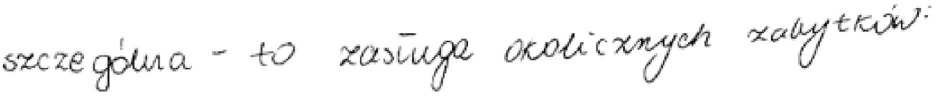
miejsce dla wszystkich turystów, którzy jeszcze

```
[AM]: ie uyin is trationie. Aemoofera just th
[BS]: ie uyin is trationie. Aemoofera just th
[D ]: like buying his tradi
```



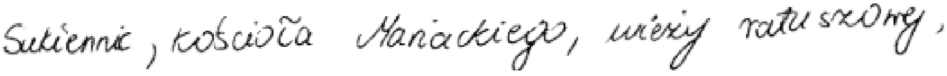
ie byli w Krakowie. Atmosfera jest tu

```
[AM]: secregama - to rasinge orolicanych rabytrcin"
[BS]: secregama - to rasinge orolicanych rabytrcin"
[D ]: secregame - to rising th
```



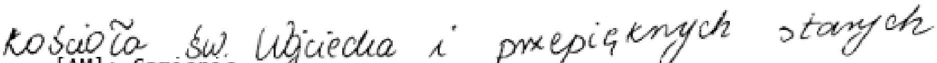
szczególna - to zasługa okolicznych zabytków:

```
[AM]: Suxiennic, robcidia Mariactiegs, wieing ratuseoney.
[BS]: Suxiennic, robcidia Mariactiegs, wieing ratuseoney.
[D ]: Susannic, solicitical Marc
```



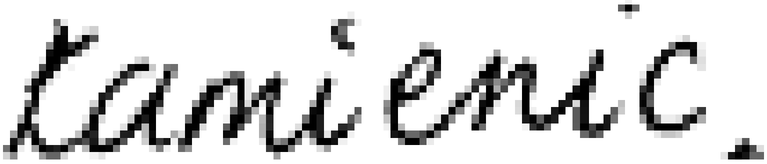
Sukiennic, kościoła Mariackiego, wieży ratuszowej,

```
[AM]: Rosccia du. Uigciecka is preepierengch stawyoh
[BS]: Rosccia du. Uigciecka is preepierengch stawyoh
[D ]: Rebeccia du. Micnicka is
```



kościoła św. Wojciecha i przepięknych stawych

```
[AM]: Camienic.
[BS]: Camienic.
[D ]: Cameric.
```



Kamienic.

[redacted]

Wyniki ilościowe

Iterujemy przez zbiór testowy, aby otrzymać wskaźnik błędu znaku (ang. Character Error Rate, CER).

[26]:
```
sclite = ScliteHelper('../SCTK/bin')

def get_qualitative_results_lines(denoise_func):
    sclite.clear()
    test_ds_line = IAMDataset("line", train=False)
    for i in tqdm(range(1, len(test_ds_line))):
        image, text = test_ds_line[i]
        line_image = exposure.adjust_gamma(image, 1)
        line_image = handwriting_recognition_transform(line_image,␣
 ↪line_image_size)
        character_probabilities = handwriting_line_recognition_net(line_image.
 ↪as_in_context(ctx))
        decoded_text = denoise_func(character_probabilities)
        actual_text = text[0].replace("&quot;", '"').replace("&apos;","'").
 ↪replace("&amp;", "&")
        sclite.add_text([decoded_text], [actual_text])

    cer, er = sclite.get_cer()
    print("Mean CER = {}".format(cer))
    return cer
```

[27]:
```
def get_qualitative_results(denoise_func):
    sclite.clear()
    for i in tqdm(range(1, len(test_ds))):
        image, text = test_ds[i]
        resized_image = paragraph_segmentation_transform(image,␣
 ↪image_size=form_size)
        paragraph_bb = paragraph_segmentation_net(resized_image.
 ↪as_in_context(ctx))
        paragraph_bb = paragraph_bb[0].asnumpy()
        paragraph_bb = expand_bounding_box(paragraph_bb, expand_bb_scale_x=0.
 ↪01,
                                               expand_bb_scale_y=0.01)
        paragraph_segmented_image = crop_handwriting_page(image, paragraph_bb,␣
 ↪image_size=segmented_paragraph_size)
        word_bb = predict_bounding_boxes(word_segmentation_net,␣
 ↪paragraph_segmented_image, min_c, overlap_thres, topk, ctx)
        line_bbs = sort_bbs_line_by_line(word_bb, y_overlap=0.4)
        line_images = crop_line_images(paragraph_segmented_image, line_bbs)

        predicted_text = []
        for line_image in line_images:
            line_image = exposure.adjust_gamma(line_image, 1)
            line_image = handwriting_recognition_transform(line_image,␣
 ↪line_image_size)
```

```
            character_probabilities =␣
 ↪handwriting_line_recognition_net(line_image.as_in_context(ctx))
            decoded_text = denoise_func(character_probabilities)
            predicted_text.append(decoded_text)

        actual_text = text[0].replace("&quot;", '"').replace("&apos;","'").
 ↪replace("&amp;", "&")
        actual_text = actual_text.split("\n")
        if len(predicted_text) > len(actual_text):
            predicted_text = predicted_text[:len(actual_text)]
        sclite.add_text(predicted_text, actual_text)

    cer, _ = sclite.get_cer()
    print("Mean CER = {}".format(cer))
    return cer
```

CER z liniami wstępnie segmentowanymi

[28]:
```
get_qualitative_results_lines(get_arg_max)
```

```
100%|                           | 1859/1859 [01:24<00:00,
22.12it/s]

Mean CER = 8.4
```

[28]: 8.4

[29]:
```
get_qualitative_results_lines(get_denoised)
```

```
100%|                           | 1859/1859 [13:18<00:00,
2.33it/s]

Mean CER = 38.3
```

[29]: 38.3

[30]:
```
get_qualitative_results(get_arg_max)
```

```
100%|                              | 3/3 [00:01<00:00,
2.27it/s]

Mean CER = 45.4
```

[30]: 45.4

[31]:
```
get_qualitative_results(get_beam_search)
```

```
100%|                              | 3/3 [00:11<00:00,
3.70s/it]
```

```
Mean CER = 45.3
```

```
[31]: 45.3
```

```
[32]: get_qualitative_results(get_denoised)
```

```
100%|                              | 3/3 [00:11<00:00,
3.70s/it]

Mean CER = 65.4
```

```
[32]: 65.4
```