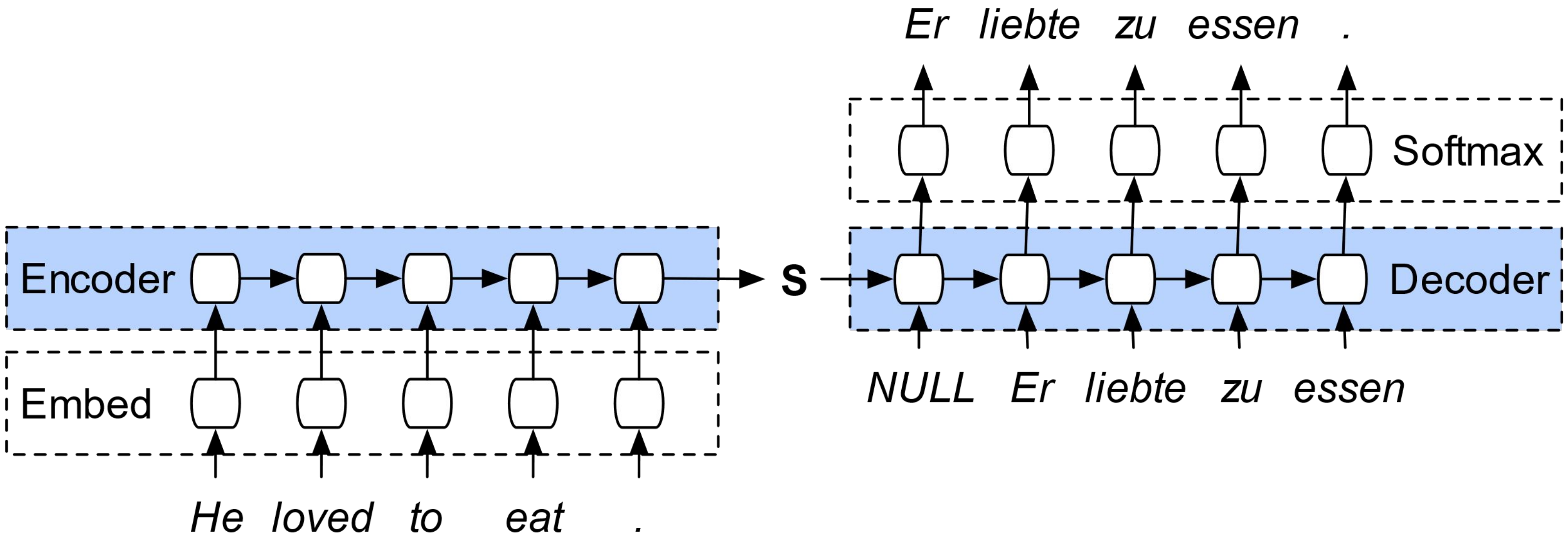
Assignment 4 Neural Machine Translation

TEAM 2
Vignesh Murali

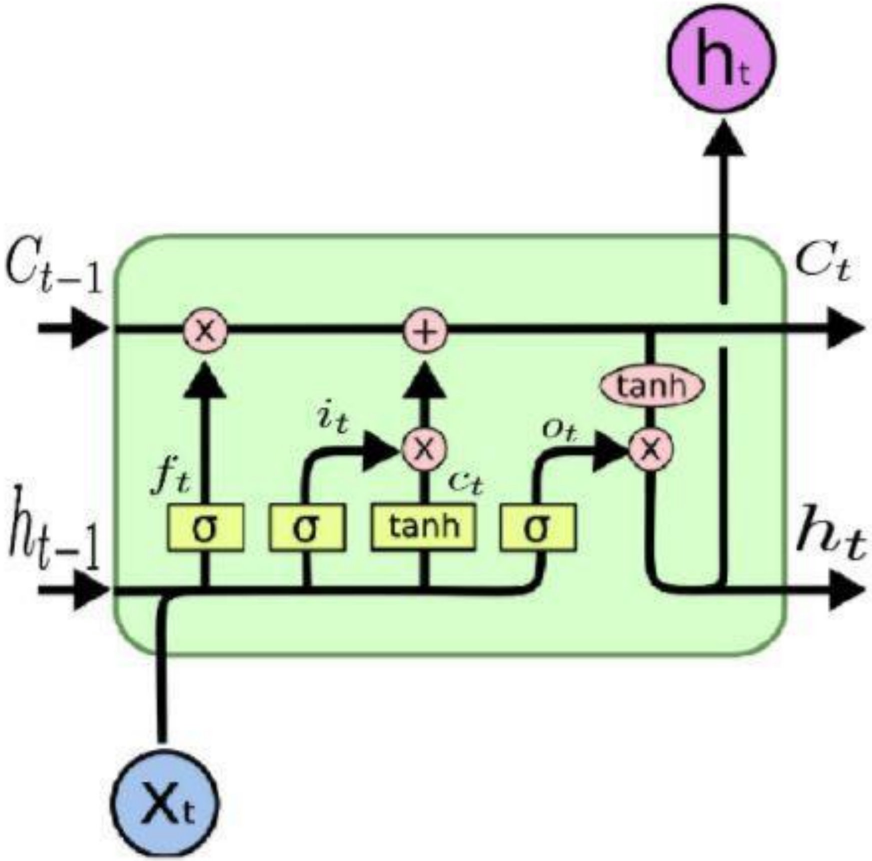
Ziqing Lu

Ritvik Reddy

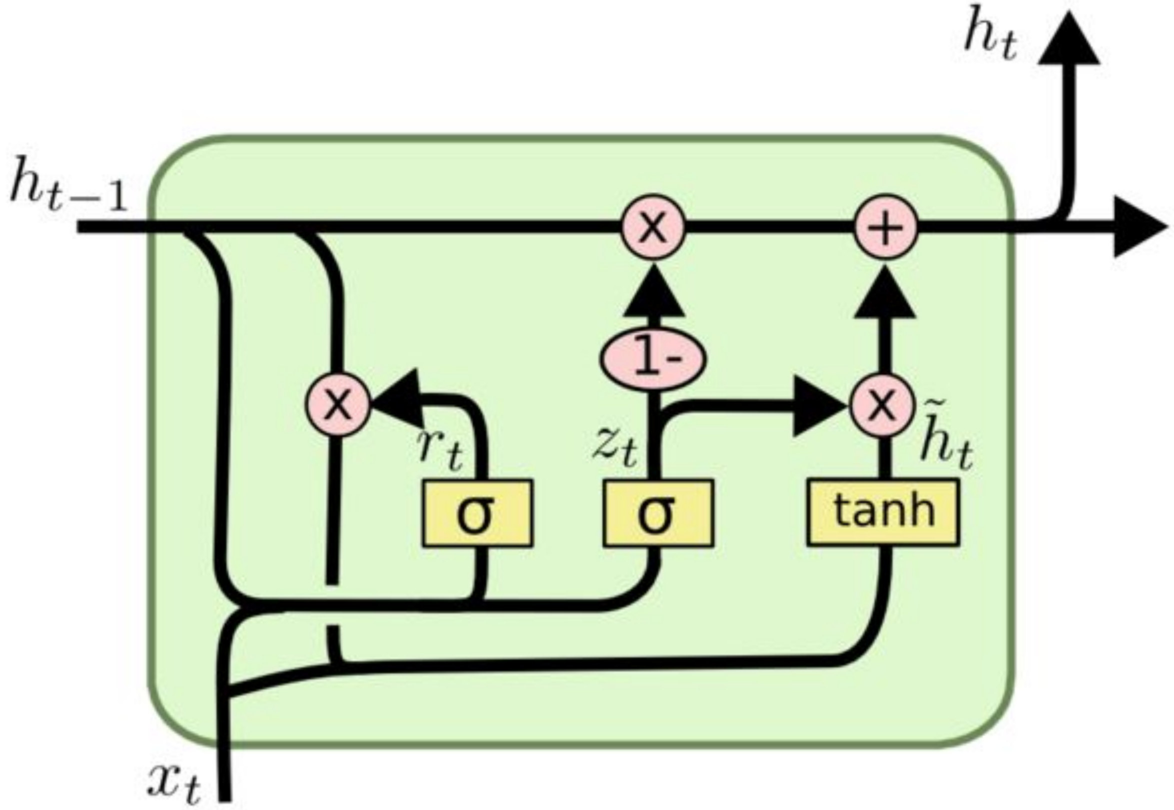
Neural Machine Translation



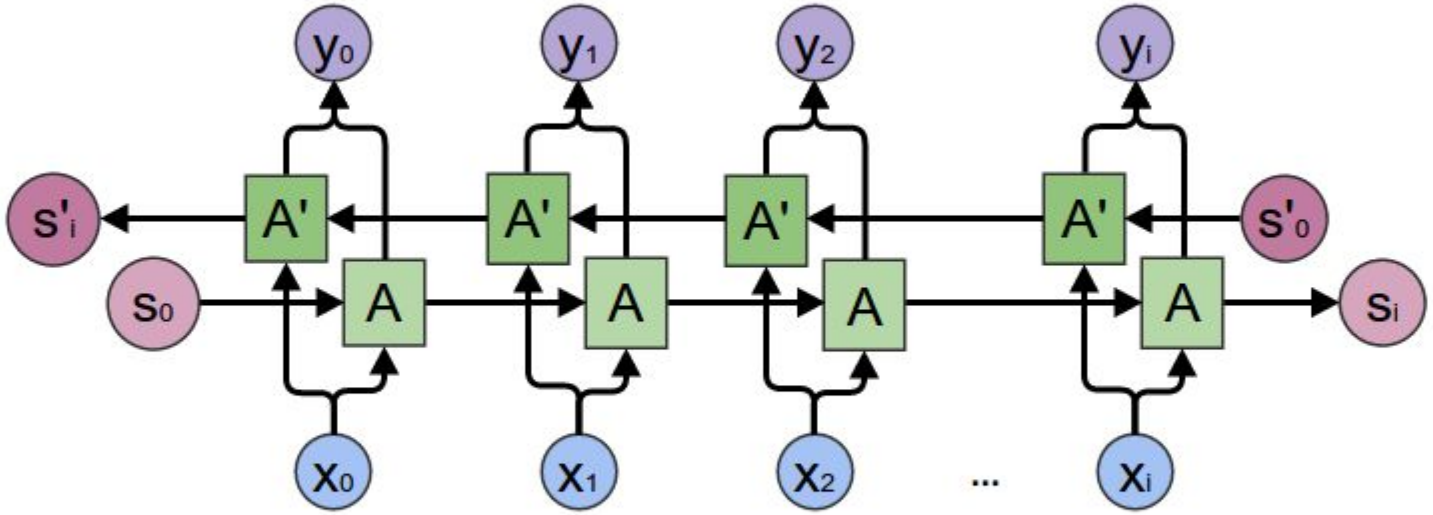
LSTM



GRU



BiDirectional RNN



Dataset

http://www.manythings.org/anki/

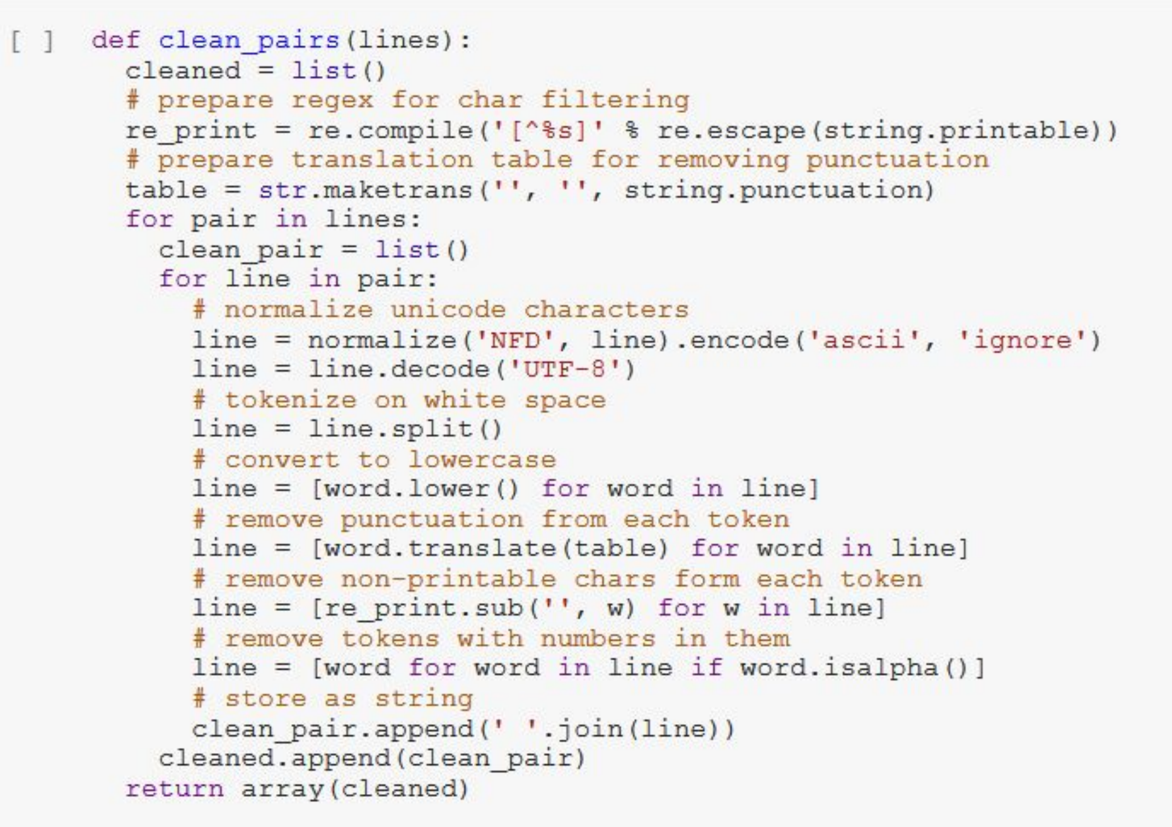
Tab-delimited Bilingual Sentence Pairs

Tom broke the window.	Tom hat das Fenster zerbrochen.
Tom checked the time.	Tom überprüfte die Zeit.

Preprocessing

We perform the following preprocessing steps:-

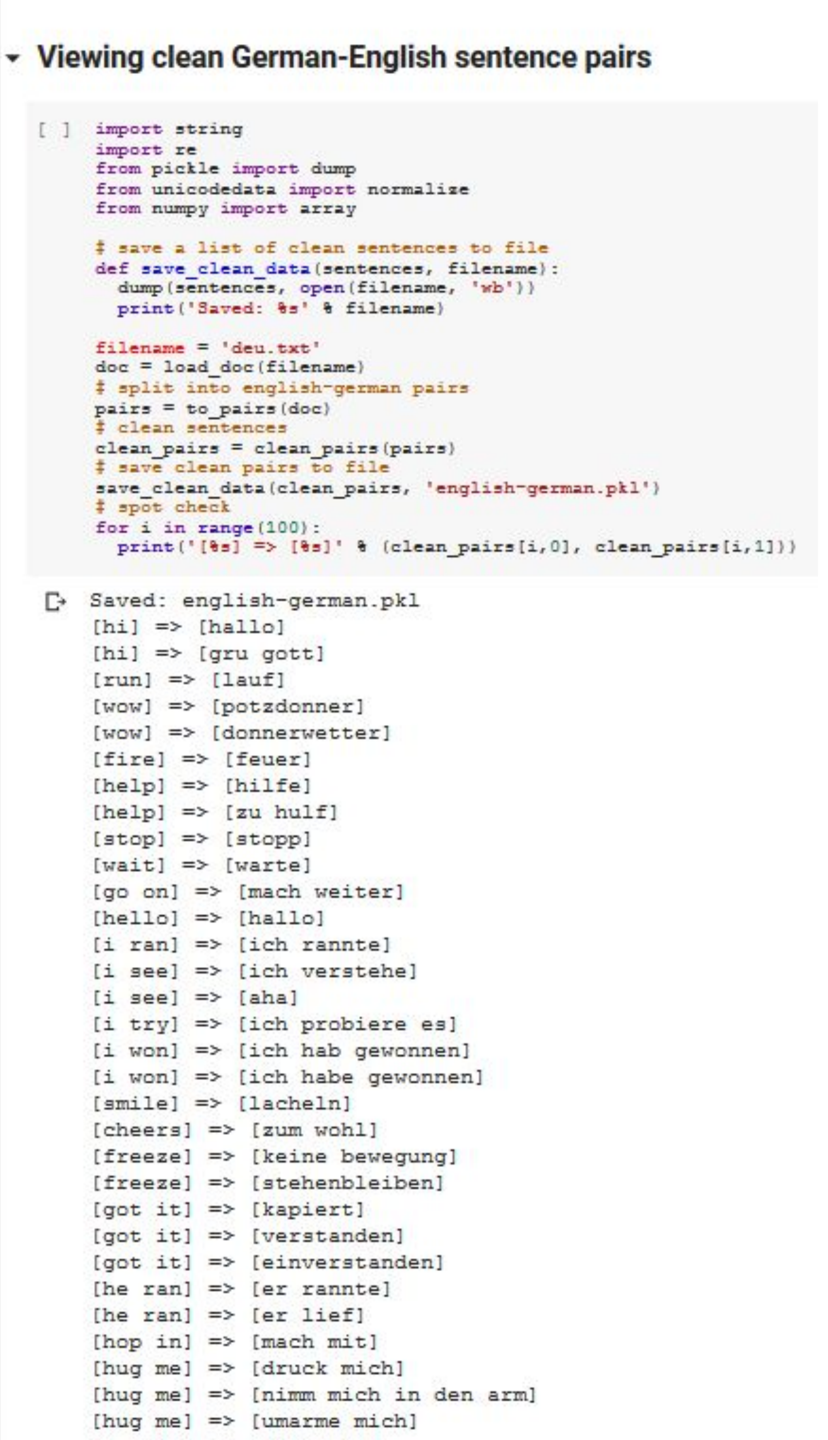
1. remove all the non-printable characters
2. remove all the punctuation characters
3. normalize all unicode characters to ASCII
4. normalize the case to lowercase
5. remove any tokens that are not alphabetic



```
def clean_pairs(lines):
  cleaned = list()
  # prepare regex for char filtering
  re_print = re.compile('[^%s]' % re.escape(string.printable))
  # prepare translation table for removing punctuation
  table = str.maketrans('', '', string.punctuation)
  for pair in lines:
    clean_pair = list()
    for line in pair:
      # normalize unicode characters
      line = normalize('NFD', line).encode('ascii', 'ignore')
      line = line.decode('UTF-8')
      # tokenize on white space
      line = line.split()
      # convert to lowercase
      line = [word.lower() for word in line]
      # remove punctuation from each token
      line = [word.translate(table) for word in line]
      # remove non-printable chars form each token
      line = [re_print.sub('', w) for w in line]
      # remove tokens with numbers in them
      line = [word for word in line if word.isalpha()]
      # store as string
      clean_pair.append(' '.join(line))
    cleaned.append(clean_pair)
  return array(cleaned)
```

Preprocessing

Use clean German-English sentence pairs

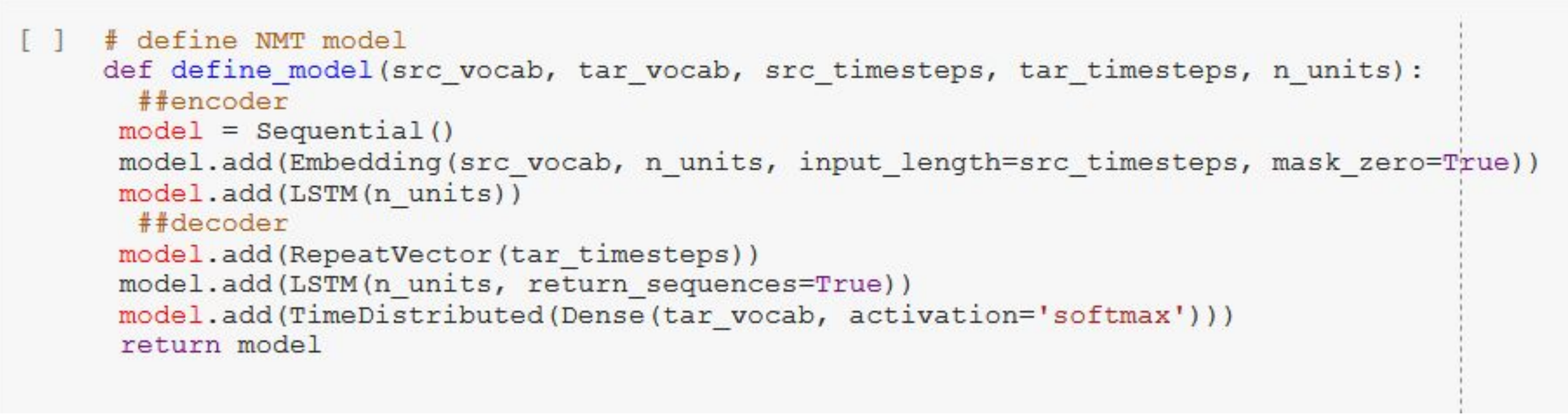


Viewing clean German-English sentence pairs

```
import string
import re
from pickle import dump
from unicodedata import normalize
from numpy import array

# save a list of clean sentences to file
def save_clean_data(sentences, filename):
  dump(sentences, open(filename, 'wb'))
  print('Saved: %s' % filename)

filename = 'deu.txt'
doc = load_doc(filename)
# split into english-german pairs
pairs = to_pairs(doc)
# clean sentences
clean_pairs = clean_pairs(pairs)
# save clean pairs to file
save_clean_data(clean_pairs, 'english-german.pkl')
# spot check
for i in range(100):
  print('[%s] => [%s]' % (clean_pairs[i,0], clean_pairs[i,1]))
```

```
Saved: english-german.pkl
[hi] => [hallo]
[hi] => [gru gott]
[run] => [lauf]
[wow] => [potzdonner]
[wow] => [donnerwetter]
[fire] => [feuer]
[help] => [hilfe]
[help] => [zu hulf]
[stop] => [stopp]
[wait] => [warte]
[go on] => [mach weiter]
[hello] => [hallo]
[i ran] => [ich rannte]
[i see] => [ich verstehe]
[i see] => [aha]
[i try] => [ich probiere es]
[i won] => [ich hab gewonnen]
[i won] => [ich habe gewonnen]
[smile] => [lacheln]
[cheers] => [zum wohl]
[freeze] => [keine bewegung]
[freeze] => [stehenbleiben]
[got it] => [kapiert]
[got it] => [verstanden]
[got it] => [einverstanden]
[he ran] => [er rannte]
[he ran] => [er lief]
[hop in] => [mach mit]
[hug me] => [druck mich]
[hug me] => [nimm mich in den arm]
[hug me] => [umarme mich]
```

Building Encoder Decoder architecture



```
[ ]  # define NMT model
     def define_model(src_vocab, tar_vocab, src_timesteps, tar_timesteps, n_units):
       ##encoder
      model = Sequential()
      model.add(Embedding(src_vocab, n_units, input_length=src_timesteps, mask_zero=True))
      model.add(LSTM(n_units))
       ##decoder
      model.add(RepeatVector(tar_timesteps))
      model.add(LSTM(n_units, return_sequences=True))
      model.add(TimeDistributed(Dense(tar_vocab, activation='softmax')))
      return model
```

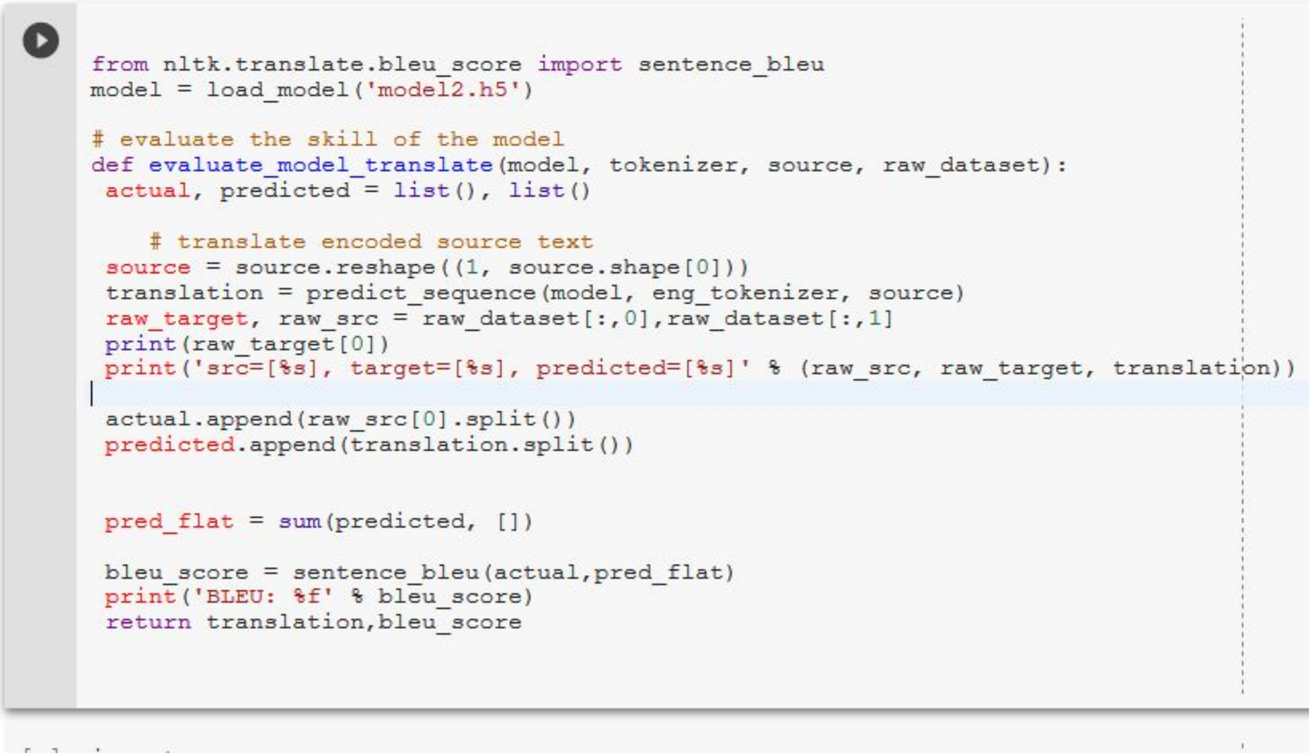
RepeatVector is used as an adapter to fit the fixed-sized 2D output of the encoder to the differing length and 3D input expected by the decoder. The TimeDistributed wrapper allows the same output layer to be reused for each element in the output sequence.

Model results

1. Single LSTM for Encoder & Decoder [Val_Acc:0.6934]
2. Single Bidirectional LSTM for Encoder & Decoder [Val_Acc:0.7298]
3. Single GRU for Encoder & Decoder [Val_Acc:0.6864]
4. Single Bidirectional GRU for Encoder & Decoder [Val_Acc:0.7194]
5. Stacked 4 LSTMs for Encoder & Decoder [Val_Acc:0.5862]

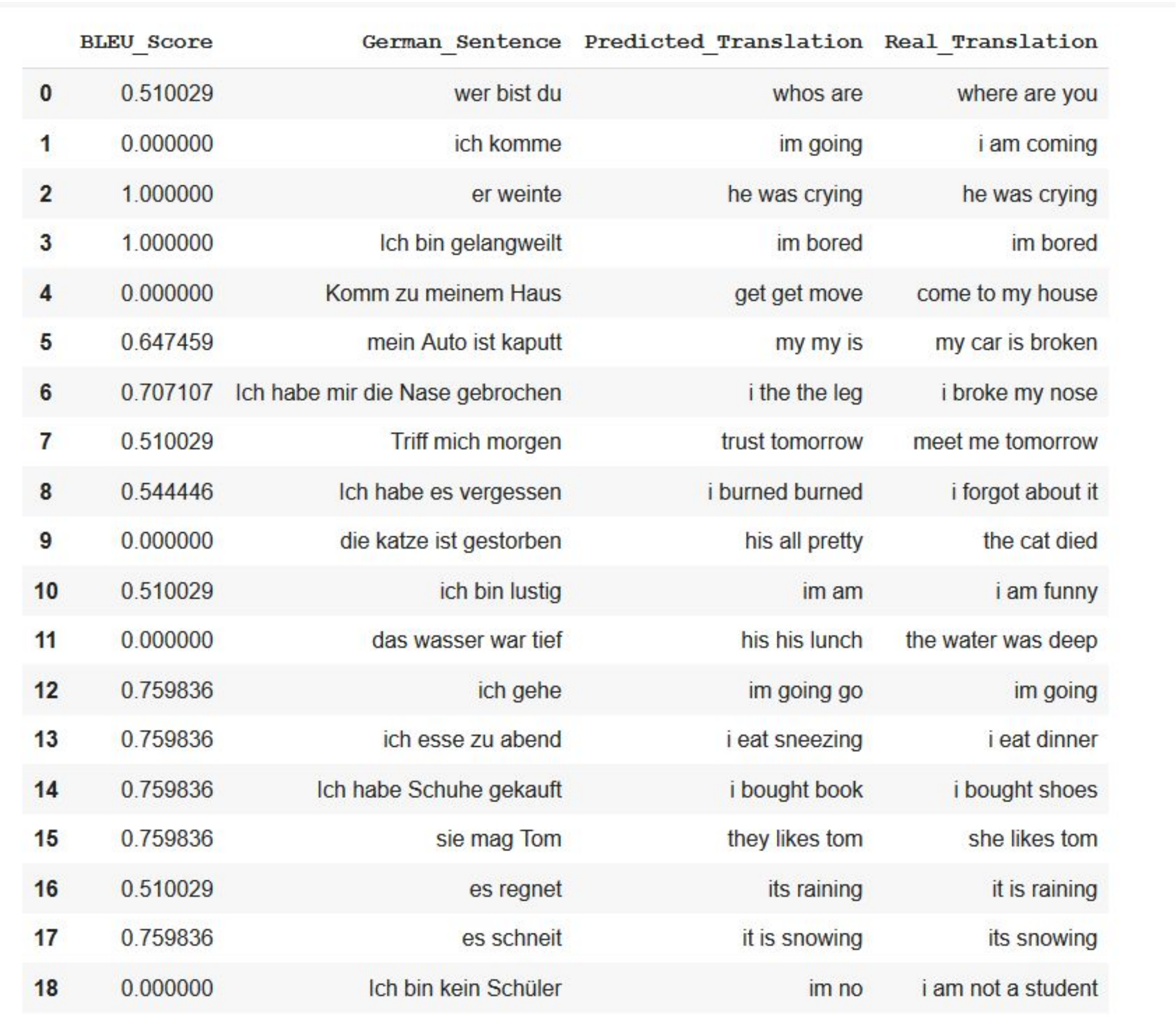
Evaluation using BLEU score

BLEU, or the Bilingual Evaluation Understudy, is a score for comparing a candidate translation of text to one or more reference translations.



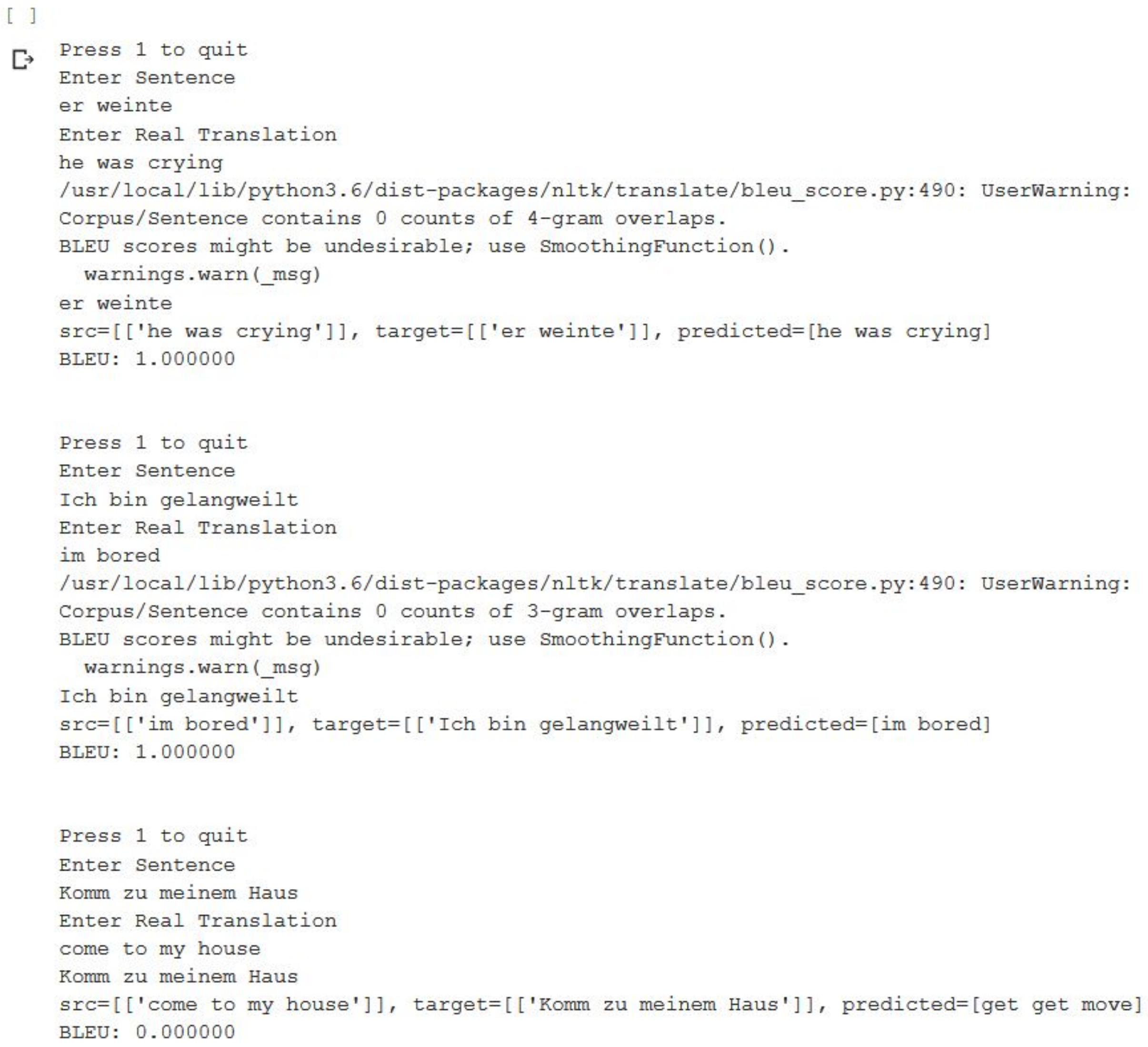
```
from nltk.translate.bleu_score import sentence_bleu
model = load_model('model2.h5')

# evaluate the skill of the model
def evaluate_model_translate(model, tokenizer, source, raw_dataset):
 actual, predicted = list(), list()

    # translate encoded source text
 source = source.reshape((1, source.shape[0]))
 translation = predict_sequence(model, eng_tokenizer, source)
 raw_target, raw_src = raw_dataset[:,0],raw_dataset[:,1]
 print(raw_target[0])
 print('src=[%s], target=[%s], predicted=[%s]' % (raw_src, raw_target, translation))

 actual.append(raw_src[0].split())
 predicted.append(translation.split())


 pred_flat = sum(predicted, [])

 bleu_score = sentence_bleu(actual,pred_flat)
 print('BLEU: %f' % bleu_score)
 return translation,bleu_score
```

Evaluation using BLEU score



	BLEU_Score	German_Sentence	Predicted_Translation	Real_Translation
0	0.510029	wer bist du	whos are	where are you
1	0.000000	ich komme	im going	i am coming
2	1.000000	er weinte	he was crying	he was crying
3	1.000000	Ich bin gelangweilt	im bored	im bored
4	0.000000	Komm zu meinem Haus	get get move	come to my house
5	0.647459	mein Auto ist kaputt	my my is	my car is broken
6	0.707107	Ich habe mir die Nase gebrochen	i the the leg	i broke my nose
7	0.510029	Triff mich morgen	trust tomorrow	meet me tomorrow
8	0.544446	Ich habe es vergessen	i burned burned	i forgot about it
9	0.000000	die katze ist gestorben	his all pretty	the cat died
10	0.510029	ich bin lustig	im am	i am funny
11	0.000000	das wasser war tief	his his lunch	the water was deep
12	0.759836	ich gehe	im going go	im going
13	0.759836	ich esse zu abend	i eat sneezing	i eat dinner
14	0.759836	Ich habe Schuhe gekauft	i bought book	i bought shoes
15	0.759836	sie mag Tom	they likes tom	she likes tom
16	0.510029	es regnet	its raining	it is raining
17	0.759836	es schneit	it is snowing	its snowing
18	0.000000	Ich bin kein Schüler	im no	i am not a student

Chatbot Demo



```
[ ]

Press 1 to quit
Enter Sentence
er weinte
Enter Real Translation
he was crying
/usr/local/lib/python3.6/dist-packages/nltk/translate/bleu_score.py:490: UserWarning:
Corpus/Sentence contains 0 counts of 4-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
er weinte
src=[['he was crying']], target=[['er weinte']], predicted=[he was crying]
BLEU: 1.000000


Press 1 to quit
Enter Sentence
Ich bin gelangweilt
Enter Real Translation
im bored
/usr/local/lib/python3.6/dist-packages/nltk/translate/bleu_score.py:490: UserWarning:
Corpus/Sentence contains 0 counts of 3-gram overlaps.
BLEU scores might be undesirable; use SmoothingFunction().
  warnings.warn(_msg)
Ich bin gelangweilt
src=[['im bored']], target=[['Ich bin gelangweilt']], predicted=[im bored]
BLEU: 1.000000


Press 1 to quit
Enter Sentence
Komm zu meinem Haus
Enter Real Translation
come to my house
Komm zu meinem Haus
src=[['come to my house']], target=[['Komm zu meinem Haus']], predicted=[get get move]
BLEU: 0.000000
```