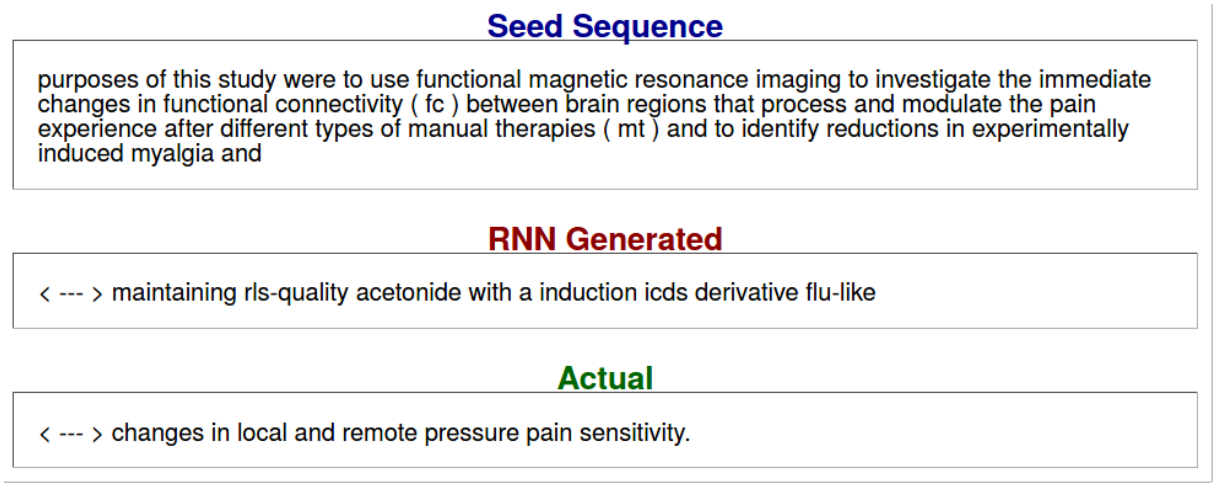
Übung 4: Long short-term memory neural networks

In this exercise you will learn how to use a specific type of a recurrent neural network, called long short-term memory (LSTM) neural network. LSTMs will be used for language modelling, and specifically, will be applied for the sentence completion task. Your goal is to train LSTMs to generate sentences starting from seed sentences, and to asses the quality of the output. Below is an example of a sequence generated by an LSTM network for a fixed input, as well as the expected completion of the input sequence. This network has been trained on the text corpus generated from PubMed abstracts. As you can see, the output of the network and the expected completion do not quite match.



Try to train a better network!

This exercise re-uses Python code for the analyzes of LSTM networks from this public notebook https://github.com/WillKoehrsen/recurrent-neural-networks/blob/master/notebooks/Quick%20Start%20to%20Rec
You are encouraged to familiarize yourself with the content presented there, which focuses on building a language model for patent abstracts.

```
%load_ext autoreload
%autoreload 2

import numpy as np
import pandas as pd
import keras
import utils
import re
from IPython.display import display, HTML
```

Get data

We have prepared a small dataset consisting of randomized controlled trials PubMed abstracts. This is a very small subset of a public dataset available here (https://github.com/Franck-Dernoncourt/pubmed-rct).

```
abstracts = []
with open('./data/processed-abstracts.txt', 'r') as f:
    for a in f.readlines():
        abstracts.append(utils.format_sequence(a))

for a in abstracts[:5]:
    print(a)
```

```
This study analyzed liver function abnormalities in heart failure patients admitted

Minimally invasive endovascular aneurysm repair ( EVAR ) could be a surgical techniq

The aim of this study was to analyse the cost-effectiveness and cost-utility of EVAR

Evidence suggests that individuals with social anxiety demonstrate vigilance to soc

Exposure to diesel exhaust causes inflammatory responses .
```

```
filters='!"%;[\\]^_`{|}~\t\n'
training_len=25
lower=False
word_idx, idx_word, num_words, word_counts, \
texts, sequences, features, labels = utils.make_sequences(
        abstracts, training_len, lower, filters)
X_train, y_train = utils.make_dataset(features, labels, num_words)
```

```
There are 15263 unique words.
There are 7142 sequences.
```

Training the LSTM model

We will train our LSTM model for 50 epochs, and analyze the quality of its ouput.

```
model_pubmed = keras.Sequential()
```

```
# Embedding layer
model_pubmed.add(
    keras.layers.Embedding(
        input_dim=len(word_idx) + 1,
        output_dim=100,
        weights=None,
        trainable=True))

# Recurrent layer
model_pubmed.add(
    keras.layers.LSTM(
        64, return_sequences=False, dropout=0.1,
        recurrent_dropout=0.1))

# Fully connected layer
model_pubmed.add(keras.layers.Dense(64, activation='relu'))

# Dropout for regularization
model_pubmed.add(keras.layers.Dropout(0.5))

# Output layer
model_pubmed.add(keras.layers.Dense(len(word_idx) + 1, activation='softmax'))

# Compile the model
model_pubmed.compile(
    optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

model_pubmed.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, None, 100)         1526300
_________________________________________________________________
lstm_2 (LSTM)                (None, 64)                42240
_________________________________________________________________
dense_3 (Dense)              (None, 64)                4160
```

```
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_4 (Dense)              (None, 15263)             992095
=================================================================
Total params: 2,564,795
Trainable params: 2,564,795
Non-trainable params: 0
_________________________________________________________________
```

```
epochs = 50
batch_size = 512
h = model_pubmed.fit(X_train, y_train, epochs = epochs, batch_size = batch_size)
```

```
/home/asan/.Virtualenvs3.6/keras/lib/python3.6/site-packages/tensorflow_core/pyt
  ”Converting sparse IndexedSlices to a dense Tensor of unknown shape. ”
```

```
Epoch 1/50
7142/7142 [==============================] - 2s 297us/step - loss: 9.6210 - accurac
Epoch 2/50
7142/7142 [==============================] - 2s 255us/step - loss: 9.1139 - accurac
Epoch 3/50
7142/7142 [==============================] - 2s 255us/step - loss: 7.1189 - accurac
Epoch 4/50
7142/7142 [==============================] - 2s 253us/step - loss: 6.6361 - accurac
Epoch 5/50
7142/7142 [==============================] - 2s 255us/step - loss: 6.4840 - accurac
Epoch 6/50
7142/7142 [==============================] - 2s 254us/step - loss: 6.4259 - accurac
Epoch 7/50
7142/7142 [==============================] - 2s 256us/step - loss: 6.3805 - accurac
Epoch 8/50
7142/7142 [==============================] - 2s 256us/step - loss: 6.3463 - accurac
Epoch 9/50
7142/7142 [==============================] - 2s 255us/step - loss: 6.3287 - accurac
Epoch 10/50
7142/7142 [==============================] - 2s 257us/step - loss: 6.2983 - accurac
Epoch 11/50
```

```
7142/7142 [==============================] - 2s 258us/step - loss: 6.2657 - accurac
Epoch 12/50
7142/7142 [==============================] - 2s 256us/step - loss: 6.2189 - accurac
Epoch 13/50
7142/7142 [==============================] - 2s 255us/step - loss: 6.1559 - accurac
Epoch 14/50
7142/7142 [==============================] - 2s 255us/step - loss: 6.1049 - accurac
Epoch 15/50
7142/7142 [==============================] - 2s 254us/step - loss: 6.0649 - accurac
Epoch 16/50
7142/7142 [==============================] - 2s 254us/step - loss: 6.0255 - accurac
Epoch 17/50
7142/7142 [==============================] - 2s 256us/step - loss: 6.0113 - accurac
Epoch 18/50
7142/7142 [==============================] - 2s 259us/step - loss: 5.9811 - accurac
Epoch 19/50
7142/7142 [==============================] - 2s 267us/step - loss: 5.9710 - accurac
Epoch 20/50
7142/7142 [==============================] - 2s 257us/step - loss: 5.9561 - accurac
Epoch 21/50
7142/7142 [==============================] - 2s 269us/step - loss: 5.9411 - accurac
Epoch 22/50
7142/7142 [==============================] - 2s 265us/step - loss: 5.9322 - accurac
Epoch 23/50
7142/7142 [==============================] - 2s 263us/step - loss: 5.9242 - accurac
Epoch 24/50
7142/7142 [==============================] - 2s 258us/step - loss: 5.9123 - accurac
Epoch 25/50
7142/7142 [==============================] - 2s 264us/step - loss: 5.9147 - accurac
Epoch 26/50
7142/7142 [==============================] - 2s 272us/step - loss: 5.9091 - accurac
Epoch 27/50
7142/7142 [==============================] - 2s 277us/step - loss: 5.8974 - accurac
Epoch 28/50
7142/7142 [==============================] - 2s 258us/step - loss: 5.8850 - accurac
Epoch 29/50
7142/7142 [==============================] - 2s 258us/step - loss: 5.8762 - accurac
Epoch 30/50
```

```
7142/7142 [==============================] - 2s 257us/step - loss: 5.8793 - accurac
Epoch 31/50
7142/7142 [==============================] - 2s 259us/step - loss: 5.8736 - accurac
Epoch 32/50
7142/7142 [==============================] - 2s 264us/step - loss: 5.8634 - accurac
Epoch 33/50
7142/7142 [==============================] - 2s 260us/step - loss: 5.8543 - accurac
Epoch 34/50
7142/7142 [==============================] - 2s 267us/step - loss: 5.8587 - accurac
Epoch 35/50
7142/7142 [==============================] - 2s 271us/step - loss: 5.8449 - accurac
Epoch 36/50
7142/7142 [==============================] - 2s 266us/step - loss: 5.8352 - accurac
Epoch 37/50
7142/7142 [==============================] - 2s 274us/step - loss: 5.8294 - accurac
Epoch 38/50
7142/7142 [==============================] - 2s 270us/step - loss: 5.8282 - accurac
Epoch 39/50
7142/7142 [==============================] - 2s 255us/step - loss: 5.8142 - accurac
Epoch 40/50
7142/7142 [==============================] - 2s 256us/step - loss: 5.8021 - accurac
Epoch 41/50
7142/7142 [==============================] - 2s 256us/step - loss: 5.7839 - accurac
Epoch 42/50
7142/7142 [==============================] - 2s 255us/step - loss: 5.7610 - accurac
Epoch 43/50
7142/7142 [==============================] - 2s 257us/step - loss: 5.7494 - accurac
Epoch 44/50
7142/7142 [==============================] - 2s 256us/step - loss: 5.7353 - accurac
Epoch 45/50
7142/7142 [==============================] - 2s 256us/step - loss: 5.7184 - accurac
Epoch 46/50
7142/7142 [==============================] - 2s 258us/step - loss: 5.7121 - accurac
Epoch 47/50
7142/7142 [==============================] - 2s 259us/step - loss: 5.6981 - accurac
Epoch 48/50
7142/7142 [==============================] - 2s 264us/step - loss: 5.6825 - accurac
Epoch 49/50
```

```
7142/7142 [==============================] - 2s 262us/step - loss: 5.6606 - accurac
Epoch 50/50
7142/7142 [==============================] - 2s 275us/step - loss: 5.6548 - accurac
```

Check the reconstruction quality of the trained network. How good is the network at predicting the completion of sentences.

```
for i in utils.generate_output(model_pubmed, sequences, idx_word,
                               seed_length = 25, new_words = 10,
                               diversity = 0.75):
    display(HTML(i))
```

Seed Sequence

This prospective, randomised, two-arm clinical trial aims to investigate the feasibility, safety and effectiveness of transarterial chemoembolisation (TACE) combined with

RNN Generated

< — > the AF to delirium) polyester abrupt mental NNS forage

Actual

< — > the endovascular implantation of an iodine- seed strand for the

Check the generalization property of the network on an unseen input sequence

```
s = "Randomized controlled trials (RCTs) are the hallmark of evidence-based medi
display(
    HTML(utils.seed_sequence(model_pubmed, s, word_idx, idx_word, diversity=0.75
)
```

Input Seed Network Output

Randomized controlled trials (RCTs) are the hallmark of evidence-based medicine and form the basis for translating research data into clinical practice. This review summarizes commonly applied designs and quality indicators of RCTs to provide guidance in interpreting and critically evaluating clinical research data. of mortality and the ER of body measured in)

You can also check the "embeddings" learned for individual words in the corpus, by analyzing nearest neighbors in the embedding space. Ideally, only similar words would cluster together, while dissimilar words would be embedded far away from each other.

```
my_embs = utils.get_embeddings(model_pubmed)
utils.find_closest('infarction', my_embs, word_idx, idx_word)
```

```
Query: infarction
```

```
Word: infarction      Cosine Similarity: 1.0
Word: musicians       Cosine Similarity: 0.6987000107765198
Word: turnover        Cosine Similarity: 0.6937999725341797
Word: dysfunction     Cosine Similarity: 0.6894000172615051
Word: haemorrhage     Cosine Similarity: 0.6880999803543091
Word: volumetric      Cosine Similarity: 0.6876000165939331
Word: hCG             Cosine Similarity: 0.6833999752998352
Word: hospitals       Cosine Similarity: 0.6829000115394592
Word: advice          Cosine Similarity: 0.6809999942779541
Word: laser           Cosine Similarity: 0.6779999732971191
```

```
utils.find_closest('relapse', my_embs, word_idx, idx_word)
```

```
Query: relapse

Word: relapse         Cosine Similarity: 1.0
Word: allergen        Cosine Similarity: 0.7519000172615051
Word: frequent        Cosine Similarity: 0.7301999926567078
Word: fiber           Cosine Similarity: 0.7296000123023987
Word: road            Cosine Similarity: 0.7279000282287598
Word: ectopic         Cosine Similarity: 0.7258999943733215
Word: ST-segment-elevation Cosine Similarity: 0.7139999866485596
Word: persistent      Cosine Similarity: 0.7050999999046326
Word: groin           Cosine Similarity: 0.6998999714851379
Word: SHM             Cosine Similarity: 0.6984000205993652
```

Task 1: Train a better LSTM network. Train a better language model

Your task will be to train a better LSTM model and assess the quality of the generated text. In particular, you can try out training a network on shorter or longer sequence lengths (`training_len`). Other hyperparameters you could try out are: batch size, and the dimension of the recurrent layer. At a minimum, you should be able to *overfit* the given dataset, i.e., there should be very little difference between the predicted and the expected completions. In your report try to analyze how the network *builds* the language model, i.e., what principle does it use to "understand" the meaning of individual words?

Bonus task: Get more samples and train an even better lanuage model

Highly encouraged, but not necessary to get the full grade

Download the PubMed-RCT dataset (https://github.com/Franck-Dernoncourt/pubmed-rct) (20k or 200k) and prepare a bigger corpus (within the limits of your machine) to train your LSTM models on. Analyze these language models! Keep in mind that, the processed dataset provided to you in this exercise contains sentences labelled with “BACKGROUND” from the PubMed-RCT20k dataset.