Artifical Intelligence

Exercise 6: Project Proposal

December 19, 2017

Team

Sebastian Bek, Marvin Klaus, Daniela Schacherer

Problem Definition

Nowadays, we constantly get more used to having AI systems ease our everyday lives. In particular in the context of electronic devices we use the available features for facilitating text-processing. This can for instance be observed on a mobile phone or search engines.
In this context, we propose the development of an text autocompletion and prediction system. For that purpose, we will build a model, suggesting possible word-endings given a prefix. In advance, it will predict the following word relative to the cursor position.

Dataset and Agent Environment

By means of collection of data (sets), we will try to design a text bot gathering multiple English text sources in diverse text files. We will use several datasets originating from e.g.:
https://nlp.stanford.edu/links/statnlp.html#Corpora

Approach

We are planning to address this task by using a machine learning approach. We will implement a neural network and train it using the selected set of training data. More specifically, we would like to use a long short-term memory (LSTM) network - a special type of recurrent neural networks (RNN) - as they are commonly used in language recognition tasks, according to the literature. Through the recurrent connections a short term memory is created for every layer of the neural network which can be interpreted as memory. The recurrence of these connections keeps the gradient from vanishing which allows learning for multiple epochs. LSTM networks maintain a constant error throughout the layers. We aim at maximizing the accuracy of the approach's results compared to predefined test chunks.

Possible Challenges:
A possible challenge arises from words, which are occurring much more frequently than others, e.g. "in, the, that", since these do not indicate any hint about possibly following words. We

have to cope with this task by finding an appropriate solution. Other than that, we would have to deal with spelling mistakes of the user. We could think of a Spell Check routine, suggesting the correctly spelled word and additionally the predicted next word.

Evaluation and expected results

Firstly, we would access the accuracy (amount of reasonable predictions) of our network over time for the training set as well as for the test set. We aim at minimizing the error between the accuracy of the training data set and the accuracy of the test dataset.
We consider the following test cases:

1. Take n chars at one single random position of a text, and check if the next word in the test-text is in our top m suggestions. We calculate an accuracy for which cases the test passed. At the moment we plan to achieve an accuracy of 50%.

2. Check the accordance of predicted words in a text-test. Cases for which tets 1 passed are considered with 100% accordance.

3. Divide a test-text into a training and evaluation part and evaluate the functionality addressing closely similar (content-based) test-texts.

Hardware

We do not think we will require any specific hardware system, yet our approach's (execution and evaluation) requirements will not be especially demanding.