# EXPOSÉ FOR A MASTER THESIS

| | |
|---|---|
| MAJOR | Computer Engineering |
| WORKING TITLE | Part-of-Speech Tagging with Neural Networks for a conversational agent |
| CANDIDATE | Müller, Andreas |
| MATRICULATION NUMBER | 333471 |
| 1ST SUPERVISOR<br>2ND SUPERVISOR | Michael, Thilo<br>Hillmann, Stefan |
| PLANNED TIMEFRAME | 6 months, November 2017 to April 2018 |

## SCIENTIFIC BACKGROUND

A part-of-speech tagger is a system which automatically assigns the part of speech to words using contextual information. Potential applications for part-of-speech taggers exist in many areas including speech recognition, speech synthesis, machine translation and information retrieval. [1]

Part-of-speech tagging for natural language processing is also used in an advisory artificial conversational agent called Alex. Alex was developed to answer questions about modules and courses at the TU Berlin [2]. The system takes the written natural language requests from the user and tries to model SQL-queries out of that. To understand the natural language queries, the system uses a Hidden Markov Model (HMM) to assign tags to each word of the query (part-of-speech tagging). This HMM tagger is trained with manually created training templates that are filled with the data in the database to be queried. The few manually created sentence-templates and the slot-filling result in training data that has largely the same structure. This often leads to unwanted result when the HMM tagger is presented with a sentence, that doesn't fit into the structure of the training templates.

## OBJECTIVES

It was shown before, that the neural network approach outperforms part-of-speech tagger that are based on statistical models (like the HMM) [3]. To address the lack of flexibility of the HMM that is currently in use, two new types of classification-models should be implemented and evaluated: Feedforward Neural Networks [1] and Recurrent Neural Networks [4].

Since the HMM is only one of many processing elements of Alex, the implementation of the neural networks should be done without having to change other components of the conversational agent. If those changes are nevertheless necessary, they should be kept minimal.

To train the neural network models, a large enough data corpus should be retrieved. Already existing logfiles of user data can be used for this purpose.

## IMPLEMENTATION

Alex was constructed as a modular system, allowing to exchange several parts with a specific functionality. Therefor the training and tagging part of the system (which are currently realized by the HMM) can be replaced by any subsystem providing a new tagger class, that implements a training and a tagging method and maintains the following format for the training data as well as the result of the tagged data:

```
# input:
training_data = [[("first", "TAG"), ("sentence", "TAG")], [("next", "TAG"), ("sentence", "TAG")]]
# output:
tagged_sentence = [("this", "TAG"), ("is", "TAG"), ("a", "TAG"), ("sentence", "TAG")]
```

To implement such a subsystem with neural network architectures, the open-source machine learning library TensorFlow can be used, which allows to create feedforward as well as recurrent neural networks in Python.

Figure 1 shows a possible structure of such a feedforward neural network. The feature vector of a given word is projected by vector representations of the surrounding words (e.g. the three preceding words) and then used as input for the neural network. With one or more hidden layers and an output layer that represents all existing POS tags, the weights (V and W) between the layers can be trained to match a given word w(t) to its most likely POS tag.

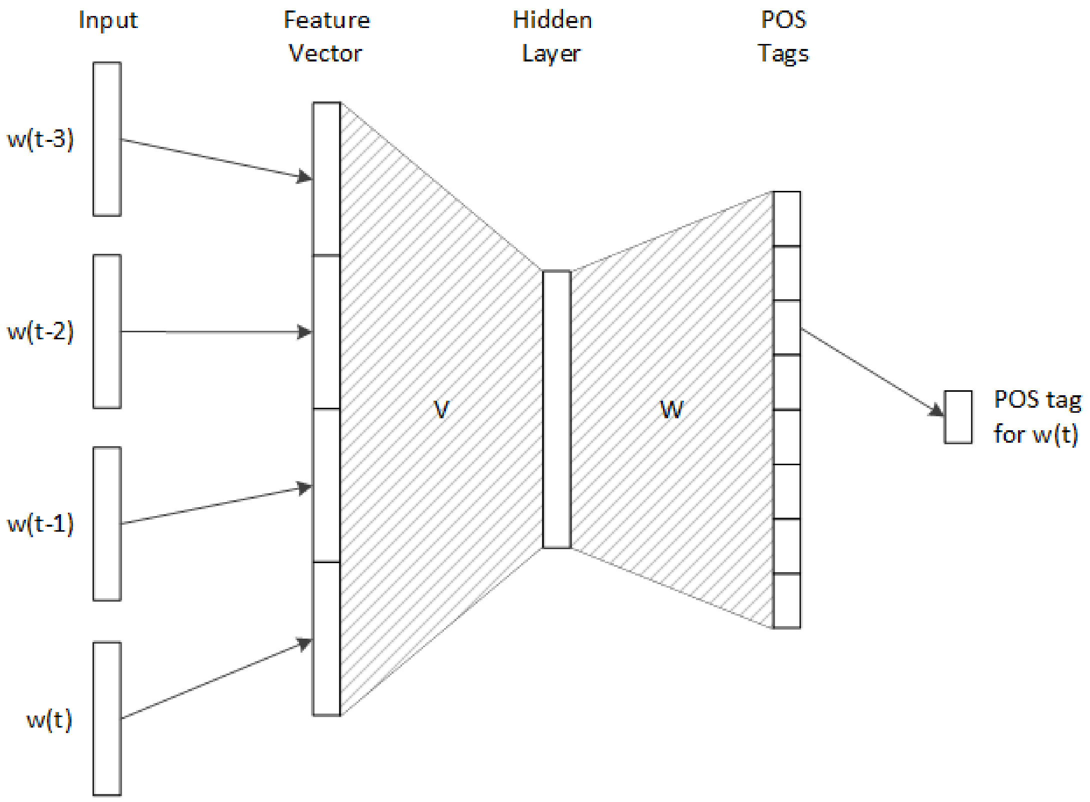


*Figure 1: Feedforward Neural Network for POS tagging*

The structure of a possible recurrent neural network is similar to the feedforward neural network, with the addition of a short-term memory. Here the weights of the training step before are taken into account for the calculation of the weights for the POS tag of the current word.

In addition to the implementation of a tagger, an evaluation system should be implemented as well to calculate a score for the quality of the trained model. The results of both neural network implementations will then be compared to those of the HMM tagger.

## SCHEDULE

The following table gives a rough overview of the different phases of this work:

| | |
|---|---|
| 2 WEEKS | Preparation and research |
| 3 – 4 WEEKS | Implementation of a Feedforward Neural Network |
| 3 – 4 WEEKS | Implementation of a Recurrent Neural Network |
| 1 – 2 WEEKS | Data retrieval and annotation for training |
| 1 – 2 WEEKS | Model training and optimization |
| 2 – 3 WEEKS | Evaluation and comparison of FNN, RNN and HMM |
| 1 WEEK | Integration into Alex |
| 6 – 8 WEEKS | Documentation and elaboration<br>(also during previous phases) |
| 19 – 26 WEEKS | |

## LITERATURE


[1] Helmut Schmid, “Part-of-speech Tagging with Neural Networks”, COLING '94 Proceedings of the 15th conference on Computational linguistics - Volume 1 Pages 172-176, Kyoto, Japan — August 05 - 09, 1994

[2] Thilo Michael, “Design and Implementation of an Advisory Artificial Conversational Agent”, 2016, Master thesis https://gitlab.tubit.tu-berlin.de/thilo.michael/aaca-alex/uploads/671cceae641fe3b8d530be9aae8a06db/thesis.pdf

[3] Q. Ma, K. Uchimoto, M. Murata and H. Isahara, "Elastic neural networks for part of speech tagging," Neural Networks, 1999. IJCNN '99. International Joint Conference on, Washington, DC, 1999, pp. 2991-2996 vol.5.

[4] J. A. Perez-Ortiz and M. L. Forcada, "Part-of-speech tagging with recurrent neural networks," Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on, Washington, DC, 2001, pp. 1588-1592 vol.3.