# Deep Learning Project Report
# Task 2: Named Entity Recognition

Samantha Tureski
4109680

Jay Kejriwal
4142919

January 31, 2018

## 1 Introduction

In this project, we sought to build a deep learning model that would accurately recognize German named entities from data in CoNLL-X format. The current version of the model is an improved attempt from our original submission.

In the pre-processing stage, we made modifications to the model we had used in the second Deep Learning assignment. In `train.py`, all of the words and their named entity tags are extracted from the data file into separate lists that correspond to sentences. To make the data more accessible to the model, elements in both the set of tags and the set of individual words are given a unique integer value using Daniel de Kok's Numberer class. The batches are then created: Numpy arrays of zeros are initialized in order to ensure that data passed into the model will have a uniform shape. Batch size is calculated depending on the number of batches set in the configuration file, `configuration.py`. Every batch contains a list of sentences with their words encoded as integers, corresponding lists of integer-encoded tags for each sentence, and a list of the lengths of each of these sentences.

The model itself uses the Tensorflow function `tf.nn.embedding_lookup` to create vector representations of each word. Once these word embeddings are determined, dropout is applied. This input layer is passed through a bidirectional recurrent neural network containing bidirectional Gated Recurrent Unit layers. We flatten the output of our bidirectional RNN to get the output of every timestep. After computing the scores of tags with Tensorflow's `tf.matmul` function, we reshape the tensors back to their original shapes. Next, we calculate the losses using `tf.nn.sparse_softmax_cross_entropy_with_logits`. During the training phase, we chose to use the Adaptive Moment Estimation (Adam) optimizer to allow our model to learn and minimize losses efficiently. Precision and recall were initially calculated using the respective functions from the `tf.metrics` module, but, after closer inspection into the Tensorflow documentation, labels and their predictions will be cast to boolean values during the calculations, suggesting that the module only currently supports binary classification tasks. As a result, we chose to use a metrics function from Scikit-learn multi-class classification problems instead, and this produced more reasonable values for

our model's performance. We have kept both sets of values in the final project to exemplify our learning process.

## 2 Results and Further Remarks

Our final model achieved an accuracy of 95.56%, a precision of 51.43%, a recall of 25.25%, and an F1 score of 30.67 after 10 epochs. These values approach the reasonable range of performance for German named entity recognition as described in the CoNLL-2003 shared task by Tjong Kim Sang and De Meulder (2003), where baseline values were 31.86% and 28.89% for precision and recall, respectively.

At the first stages of our working model, we had achieved results of 56.58% accuracy, precision of 61.01%, recall of 100.00%, and an F1 score of 75.78, after 10 epochs. The extraordinarily high recall value gave us a clue that something was off; it was dubious that the model had predicted a certain label every single time it should have been predicted. To test what had gone awry with these calculations, we inspected the predicted labels and compared them manually against the gold-standard labels. It was clear from this process that the model was learning over time, but that the recall was not perfect. When we applied the Scikit-learn metrics function for multi-class classification problems (as named entity recognition is), the values became much more reasonable.

We achieved the higher accuracy of 95.56% due to fixing a small error in the model. A higher accuracy is more plausible for this task: a properly trained model on named entity recognition should have an abnormally high accuracy measure due to most tokens being labelled "not an entity," as most texts, indeed, consist of mostly common nouns.

We are pleased to say that this model performs sufficiently well. We are grateful to have been given an opportunity to correct the flaws in our old model and gain deeper insights into the process of building a neural network.

We would like to thank our professor, Daniel de Kok, for guiding us in a series of emails and for being there to answer our questions.

## References

[1] Guillaume Genthial
Sequence Tagging with Tensorflow
`https://guillaumegenthial.github.io/sequence-tagging-with-tensorflow.html`

[2] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition.