**Samtla-Char-NER Report**

**Implementation of Character-based Named Entity Recognition into the Samtla System**

*Student: Matthew Ralph,*

*MSc Computer Science project report, Department of Computer Science and Information, Birkbeck College University of London*

[*mralph02@dcs.bbk.ac.uk*](mailto:mralph02@dcs.bbk.ac.uk)

*Supervisor: Dr Dell Zhang*

*With thanks to Dr Martyn Harris*

*This report is substantially the result of my own work, expressed in my own words, except where explicitly indicated in the text. I give my permission for it to be submitted to the JISC Plagiarism Detection Service. The report may be freely copied and distributed provided the source is explicitly acknowledged.*

# Abstract

Recent approaches to Named Entity Recognition, such as that of (Kuru, Arkan Can and Deniz, 2016), demonstrate that a character-level representation of textual data can yield good results when training a deep learning. In this project, a set of Hansard debates is aggregated, processed and labelled for use in a Bidirectional Long Short-Term Memory neural network. The trained model, and the original dataset, is integrated with Birkbeck’s Samtla digital humanities text archiving system, such that the Hansard texts can be browsed in the interface, and previously unseen Named Entities are highlighted.

# Acknowledgements

I would like to thank the people who, taught me to program in Python by working on real problems: to Ali Lotia and Ogonna Iwunze, whose expertise is matched only by their patience. To Sergio Gutierrez-Santos, whose instruction in the Java programming language was well-structured and helped to open up a world of more structured code.

I am grateful to Dr Martyn Harris for his help and encouragement when exploring this project and its potential integration with Samtla, and to Dr Dell Zhang for his ideas, advice on the academic landscape surrounding Named Entity Recognition, and quick responses to my queries.

Finally, I would like to thank my wife for all her support throughout this Master’s. AMDG.

# List of figures and tables

Compile at end

Table of Contents

[0](#_Toc521058037)

[1. Abstract 1](#_Toc521058038)

[2. Acknowledgements 1](#_Toc521058039)

[3. List of figures and tables 1](#_Toc521058040)

[5. Introduction (including background) 4](#_Toc521058041)

[6. Overall Results (trailer) 4](#_Toc521058042)

[7. Software Architecture 4](#_Toc521058043)

[7.1. The Pipeline of tasks and Invoke 4](#_Toc521058044)

[7.2. Named Entity Downloading 6](#_Toc521058045)

[7.3. Raw Hansard downloading 6](#_Toc521058046)

[7.4. Hansard processing 6](#_Toc521058047)

[7.5. Hansard interpolation 6](#_Toc521058048)

[7.6. Formation of Tensors 6](#_Toc521058049)

[7.7. Overview of files in project and what they do 6](#_Toc521058050)

[8. Implementation issues 6](#_Toc521058051)

[8.1. Wikipedia data cleanliness 6](#_Toc521058052)

[8.2. TWFY API suspect return values 6](#_Toc521058053)

[8.3. NLTK span\_tokenize bugs 6](#_Toc521058054)

[8.4. Toy dataset model – tensor sparsity 6](#_Toc521058055)

[8.5. Hansard Presentation issues 6](#_Toc521058056)

[9. Testing 6](#_Toc521058057)

[9.1. Unit testing 6](#_Toc521058058)

[9.2. Manual evaluation 6](#_Toc521058059)

[9.3. Model cross-validation 6](#_Toc521058060)

[9.4. Overall evaluation 6](#_Toc521058061)

[10. Summary and Conclusions 6](#_Toc521058062)

[10.1. Pre-processing is hard 6](#_Toc521058063)

[10.2. Labelling is hard 6](#_Toc521058064)

[10.3. Sentence tokenization is hard 6](#_Toc521058065)

[11. References 6](#_Toc521058066)

[12. User Manual 6](#_Toc521058067)

[13. Appendix: Code 6](#_Toc521058068)

[14. What’s My Work 6](#_Toc521058069)

# Introduction (including background)

The brief for this project was to demonstrate Named Entity Recognition, using the approach cited in (Kuru, Arkan Can and Deniz, 2016), and the Keras implementation of this provided by GitHub user 0xnurl.[[1]](#footnote-1) The target dataset was the Hansard, the record of debates in both of the houses of Parliament in the United Kingdom.[[2]](#footnote-2) This dataset is now available via the Parliament UK Data API,[[3]](#footnote-3) however this API is largely undocumented and was not available at the start of this project.

# Overall Results (trailer)

# Software Architecture

## The Pipeline of tasks and Invoke

This project was, in essence, a data pipeline. Data was sourced from Hansard debates and form Named Entities, combined using a variety of algorithms, and then stored in a format that could then be used to predict unseen named entities. As such, it is best visualised using a pipeline flow (see Figure 1).



Figure pipeline data processing model

## Named Entity Downloading

## Raw Hansard downloading

## Hansard processing

## Hansard interpolation

## Formation of Tensors

## Overview of files in project and what they do

# Implementation issues

## Wikipedia data cleanliness

## TWFY API suspect return values

## NLTK span\_tokenize bugs

## Toy dataset model – tensor sparsity

## Hansard Presentation issues

E.g. No speaker information due to XML processing

# Testing

## Unit testing

## Manual evaluation

## Model cross-validation

## Overall evaluation

# Summary and Conclusions

## Pre-processing is hard

## Labelling is hard

## Sentence tokenization is hard

Taught specific abbreviations to the tokenizer. Still bugs outstanding.

# References

# User Manual

# Appendix: Code

# What’s My Work

1. https://github.com/0xnurl/keras\_character\_based\_ner [↑](#footnote-ref-1)
2. https://hansard.parliament.uk/ [↑](#footnote-ref-2)
3. <http://www.data.parliament.uk/dataset/12> and http://api.data.parliament.uk/ [↑](#footnote-ref-3)