



Image from [deepAI image generator](#) based on prompt: "Eye-movement based classification of unknown words with LSTMs and CNNs"

# Eye movement based classification of unknown words with RNNs

**William LaCroix**

wila00001@stud.uni-saarland.de

Matrikelnummer: 7038732

Saarland University

Winter Semester 2023-2024

## Overview

For this project I will be extending the previous research conducted by Ryzhova et al. 2023 by developing neural models for unknown word classification based on eye movement measures.

## Main Goal

1. Compare effectiveness of neural model architectures, namely LSTM and CNN, against the baseline logistic regression classifier model from Ryzhova et al. 2023.

## Motivation

In the original research, a logistic regression classifier model was trained on filler words to predict control vs pseudo words in isolation. This approach examines eye-movement measures on a per-word basis, but ignores the sequential nature of the input data. As illustrated in studies such as Haller et al. 2022, the sequential nature of human eye-movement during reading is more intuitively captured by sequential neural models such as LSTMs and CNNs.

## Hypothesis / Research Question

The main research question to be answered by this experiment is: *"does incorporating the sequential nature of the input data more accurately account for eye-movement during reading?"* This hypothesis will be supported if the predictive performance (primarily F1) of sequential neural models exceeds that of the baseline logistic regression model.

## Considerations

### I. Input features

The original study included subject specific eye-movement measures (first-duration fixation, first-pass duration, dwell time, number of fixations, and number of regressions out), as well as subject independent word features (word length, log word frequency, log trial order) as predictors. It is often important to reduce input features for regression models, to prevent overfitting and increase interpretability, but since neural models are black boxes by nature, it seems reasonable to take an

“kitchen-sink” approach to model input; including as many input features as are available in the original dataset.

## II. Modeling approach

The primary objective of this classifier is to identify/label the pseudowords in the dataset based on eye-movement measures, making this a binary classification challenge {1: known, 0: unknown}, with two (or more) possible approaches.

1. *Sentence-level prediction*: take the input features of the entire sentence, and attempt to predict whether the target word was **control | pseudo**, giving a {0,1} output per sentence. This may be an “easier” task for the neural models, but will suffer from the relatively small size of the dataset.
2. *Word-level prediction*: take input features on a word-by-word basis, and attempt to classify the words as **known | unknown**. This has the advantage of expanding the training data size from  $m$  sentences, to  $m \times n$  words, and can be viewed similarly to a machine translation task with a very limited target language vocabulary, or named entity recognition task with only 2 labels {known, unknown}.


## III. Limitations

One limiting factor to this approach will be dataset size, since neural models will have a higher tendency to simply memorize the training dataset, while generalizing poorly to the test set.

Additionally, this dataset is quite imbalanced in terms of pseudowords: in a sentence-level prediction setting, only 3 of 28 sentences are in the pseudo condition, giving any model an 89% accuracy score simply by predicting “known” for all cases. On word-level prediction, this problem is exacerbated by the nature of the input sentences, now instead of 3/28 pseudoword-containing sentences,, we have  $3/28n$  pseudowords, which could skew the model toward not making any interesting predictions.

## Contributions

This follow-up experiment will accomplish at least 2 research tasks: first, it will assess the validity and performance of sequential neural models in predicting unknown words. Secondly, there is increased opportunity for the model to identify false positive patterns from the original dataset, since the regression model looks only at **control | pseudo** pairs in testing, while a word-level predictor is attempting to classify *every* token as **known | unknown**, which may identify instances of word reading difficulty, outside the scope of the initial classification model. This has the potential to allow the model to



generalize more broadly to non-pseudo words that are nevertheless unknown to the reader, and may potentially reduce the number of false negative predictions due to unusual eye movements on filler words in testing.