# Domain Entity Recognition

**Tanishq Chaudhary**
**Mayank Goel**
**Jerrin John Thomas**
**Mentor: Hema Ala**

## Overview

The aim is to create different models for NER in the domain of sports. The major element of our project is to create a highly accurate NER for real-world data on Sports. This has high usage in fields of sports marketing, news and user experience in fields of consuming news/data about their favourite sports event.
To give an insight into different NER models, we would use FIFA World Cup 2018 as an event to form ideas on. FIFA World Cup 2018 was chosen as it is a highly popular event and that would mean tweets would be readily available.
We plan to compare a Deep Learning Approach, a CRF approach, a statistical data-independent approach.
The reason for working on multiple approaches was to be able to compare ideas and include cross-domain ideas. State-of-the-Art NERs exist, and for such events a string matching algorithm like Karp-Rabin's Algorithm based on hashing is used.
However for k words with a combined length n where $|k| = 0.01(n)$ (Assuming about one NER per sentence) we can gain a speed boost by tagging NER on each tweet itself and comparing the possible NER to the database. This is considering one NER per sentence, however most tweets do not have NERs and comparing such a long database is a faulted approach. We intend to work on fixing this problem and its potential value in other fields, focusing on the research aspect instead of trying to desperately increase accuracy of a model, most of which are heavily data-dependent.

Github Repository

## References & Data Sources

### *Papers*

- End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF
- Neural Architectures for Named Entity Recognition

## ***Dataset***

- Tweets: FIFA World Cup 2018 Tweets
- Nations: FIFA WORLD CUP 2018 Players
- Clubs, Players: FIFA World Cup
- GMB extract: Annotated Corpus for Named Entity Recognition

# Specifications

We used a dataset online which had 530k tweets collected on FIFA World Cup 2018. This was raw data that we cleaned, and then annotated. We POS tagged each sentence using NLTK, and tagged the NERs in them. The tags assigned were - Beginning and Middle Player NER, and Beginning and Middle Team NER (for purpose of multi word NERs)., using the IOB format.
The data was cleaned using the csv module in Python, and tagged using DataFrames in Pandas. The code is well-commented and is in the form of a Notebook on Github. We used binary search to speed up our annotation by a considerable degree.
We are also using the GMB extract on Kaggle which was already cleaned and put in the required format. It has many more NE tags, we converted the relevant.

## ***Deep Learning***

We have used scikit learn to create a memory tagger and tensorflow to create, compile and train the neural network.
The data was successfully cleaned and put in the required format. We now start with data exploration. The notebook graphs give a general idea of the length of the data being used, the number of sentences, the number of words for both the corpora and the number of tags: both POS tags and the NE tags.
Now, we converted the word to arbitrary integers, which are then used for encoding generation. This creates our training and testing data. At the same time we also pad the sentences to be of equal length to facilitate their conversion to tensors.
Memory Tagging: What would happen if we simply memorized all the tags?
The memory tagger simply remembers the most popular tag that any given word has. This helps us create the baseline accuracy. We train it on the tweets data. Testing on the GMB data, it gives us 92% accuracy.
Now, we finally come to the deep learning model.
We have used a bidirectional long short term memory model with time distributed layer as an output to accurately map each input sequence to the corresponding output sequence. The inputs to this model are 64 dimensional vectors output by the embedding layer.

# Results

There were issues with overfitting on the validation and testing data. On changing the validation data to be the GMB data extract, we see that our model is able to give 97% accuracy at the least. This is much better than just simply memorizing the most popular tags for the words. More robust conclusions can be drawn after completing work with other models.

# Expected Work

We expect to be finished with all three models and compare accuracy for each and present in a visually comprehensive format.
If time permits, we wish to run this model on different sports and compare accuracy.