CS224W Project Milestone: Soccer Action Prediction

Joseph Bailey
Department of Physics
Stanford University
Stanford, CA 94305
josephrb@stanford.edu

Yi Qin
Intuit Inc.
7535 Torrey Santa Fe Rd.
San Diego, CA 92129
yqin604@stanford.edu

1 Application Domain

1.1 Task Description

A professional match of soccer is inherently a highly connected and relational system where players and actions are interconnected — a perfect candidate for a graph machine learning. Our primary task is **sequential action prediction**. We aim to develop a Graph Neural Network (GNN) model that can, given the current state of play, predict the probability distribution of the next action.

Formally, we can represent a soccer match by its time-ordered sequence of actions, $\{a_1, a_2, \ldots, a_m\}$, where each a_i has an action type t_i and an associated feature vector x_i. Given a sequence of actions $\{a_1, a_2, \ldots, a_i\}$, our task is to predict the type of the next action t_{i+1}.

1.2 Dataset

We will use this Soccer Match Event Dataset, publicly available on Kaggle, generated from this larger dataset with the socceraction package. Compared to the larger dataset, the Kaggle dataset standardizes actions into the **S**occer **P**lay **A**ction **D**ata **L**anguage (**SPADL**) format.

The dataset contains approximately 3 million actions from 1,941 matches across seven major competitions (five national European competitions in the 2017/2018 season, the World Cup 2018, and European Cup 2016), providing sufficient data for a neural network approach. Each on-the-ball action in the SPADL format is defined by twelve attributes, including:

- `type_name` (e.g., pass, dribble, shot)
- `result_name` (success or fail)
- Spatial data: `start_x`, `start_y`, `end_x`, `end_y`
- Contextual data: `time_seconds`, `team_id`, `player_id`, `bodypart_id`

1.2.1 Evaluation Metrics and Real-World Implications

Success will be measured by the model's ability to predict the **most productive next action**, where "productivity" is defined by its contribution to the likelihood of scoring a goal.

We will use two primary metrics:

- **Top-k accuracy:** the percentage of times the correct next action appears within the top k predictions.
- **Macro F1-score:** a class-averaged measure of precision and recall, important because our dataset has class imbalance (passes are more common than shots).

The real-world implications include:

Preprint. Under review.

- **Tactical Coaching:** Providing quantitative insight to coaches to identify high-value decision points and specific suboptimal sequences.
- **Advanced Scouting:** Offering a context-aware measure of a player's decision-making quality, moving beyond simple statistics.

2 Graph Attention Network

We first consider a Graph Attention Network (GAT) architecture. GAT's attention mechanism allows it to assign higher attention scores to the nearby actions that are most relevant, distinguishing high-impact actions from low-value actions.

2.1 Graph Construction

For each match, we construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each action a_i is represented by a node $v_i \in \mathcal{V}$. Edges $\mathcal{E}$ between nodes are added based on two criteria:

- An edge (v_i, v_j) is created if the two actions a_i and a_j occur closely in space and in time. That is, if a_i and a_j have start positions and times of (x_i, y_i, t_i) and (x_j, y_j, t_j) respectively, then we add (v_i, v_j) to $\mathcal{E}$ if both

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < \tau_s \qquad \text{and} \qquad |t_i - t_j| < \tau_t,$$

  where τ_s and τ_t are hyperparameters initially set to 10 meters and 5 seconds.
- An edge (v_i, v_j) is also created if actions a_i and a_j occur directly before/after one another.

This should preserve the relationship that actions that happen nearby each other are closely related. Each node v_i is initialized with an 11-dimensional embedding of normalized features:

$$h_i^{(0)} = [x_{\text{start}}, y_{\text{start}}, x_{\text{end}}, y_{\text{end}}, \Delta x, \Delta y, \text{dist}, t, \text{result}, \text{bodypart}, \text{period}].$$

2.2 Model Architecture

Our model consists of two GAT layers [Veličković et al., 2018], learning node embeddings by learning attention scores between neighboring nodes. For each layer, we use 8 attention heads with skip connections and batch normalization. The update rules are

$$h^{(1)} = \text{ELU}\left(\text{BN}\left(\text{GAT}_1(h^{(0)}, \mathcal{E})\right) + Wh^{(0)}\right)$$
$$h^{(2)} = \text{ELU}\left(\text{BN}\left(\text{GAT}_2(h^{(1)}, \mathcal{E})\right) + h^{(1)}\right),$$

where W is a learnable projection matrix used to match dimensions. The final node embeddings $h^{(2)}$ are passed through a linear output layer to convert the node embeddings to the action types.

2.3 Preliminary Results

From our dataset, we constructed 1,941 game graphs. The data was randomly split into 70% training, 15% validation, and 15% test sets. We trained for 60 epochs with Adam and a mini-batch size of 8, meaning each training batch contained 8 game graphs. To prevent overfitting, we trained with dropout after each GAT layer. Our loss function was unweighted cross-entropy loss, and we evaluated our model with accuracy, Top-3 accuracy, and Macro F1-score. The 36th epoch was best, reporting a validation accuracy of 0.682, Top-3 accuracy of 0.865, and Macro F1 of 0.107.

This is a relatively poor Macro F1-score, which can be explained in the fact that the model is great at predicting the most common action, pass, with F1 = 0.813, but not as good at predicting most other classes (see 1). The implementation and output plots can be seen here.

We also attempted to use weighted cross-entropy loss, which increased the Macro F1-score to 0.120, but decreased accuracy to 0.255 and Top-3 accuracy to 0.449.

3 Graph Transformers with Self-Attention

Overview. We implement a graph-transformer proof-of-concept for next-action prediction on a single-node Action graph with temporal-aware subgraph sampling and attention biases from edge/path features.

(1) Data processing and graph construction

- **Nodes:** one `Action` node per event row (SPADL). Node features: token embedding of `type_name`; positional encoding from Laplacian eigenvectors on the Action subgraph.
- **Edges:** temporal relation $(\text{Action}, \texttt{followedBy}, \text{Action})$. Each edge carries features
$$b_{ij} = [\Delta t_{ij},\ \Delta x_{ij},\ \Delta y_{ij},\ \text{dist}_{ij},\ \text{angle}_{ij}] \in \mathbb{R}^5.$$
- **Temporal-aware k-hop sampling:** For a seed Action u with time t_u, we build the k-hop neighborhood using only Actions v with $t_v \le t_u$.

Succinct structure (Action-only used at train time):
$$\mathcal{G} = (\mathcal{V}_{\text{Action}},\ \mathcal{E}_{\text{followedBy}}), \quad |\mathcal{V}_{\text{Action}}| = N,\ \mathcal{E}_{\text{followedBy}} \subseteq \mathcal{V}_{\text{Action}} \times \mathcal{V}_{\text{Action}}.$$

(2) Message passing, attention, loss, update

Node input: $x_i = \text{Emb}_{\text{type}}(\text{type}_i) + W_p\,\text{PE}_{\text{Lap}}(i) \in \mathbb{R}^d.$

QKV: $q_i = W_q x_i, \quad k_j = W_k x_j, \quad v_j = W_v x_j.$

Edge/Path bias:
$$c_{ij} = w^\top \tilde{b}_{ij}, \quad \tilde{b}_{ij} = \begin{cases} b_{ij} & (i,j) \in \mathcal{E}, \\ \sum_{(p,q)\in \text{shortest path}(i\mapsto j)} b_{pq} & \text{if path exists in } \le K \text{ hops}, \\ 0 & \text{otherwise.} \end{cases}$$

Attention logits: $s_{ij} = \dfrac{q_i^\top k_j}{\sqrt{d_k}} + c_{ij}.$

Weights: $\alpha_{ij} = \text{softmax}_j(s_{ij}).$

Message: $z_i = \sum_j \alpha_{ij} v_j.$

Update (pre-norm block): $h_i = x_i + \text{MHA}(x;\, c), \qquad y_i = h_i + \text{MLP}(\text{LayerNorm}(h_i)).$

Loss (type-only, seeds $\mathcal{S}$ in each batch):
$$\mathcal{L} = \frac{1}{|\mathcal{S}|} \sum_{i\in\mathcal{S}} \text{CE}\big(\text{logits}_i^{(\text{type})},\ y_i^{(\text{type})}\big).$$

(3) Result (10 epochs) Training loss decreases from 0.809 to 0.267; validation accuracy rises from 0.822 to 0.909. Macro-F1 is 0.834. Frequent classes (e.g., `pass`, `cross`, `shot`) are strong; extremely rare classes (e.g., `bad_touch`, support $= 2$) are unreliable.

(4) Further improvements we can do Results indicate label imbalance (e.g., `pass` dominates). Future work includes addressing class skew and enriching graph context. In particular, we plan to: (i) incorporate heterogeneous nodes (Players/Teams) and relation types, and (ii) add node-type embeddings so multi-type self-attention can leverage richer structure.

(5) Code and output visualization Please see code and output here

4 Relational Deep Learning and RelBench

RelBench [Robinson et al., 2024] has shown state-of-the-art performance on prediction tasks over relational databases. One potentially successful idea is to extend this approach to our dataset. Their approach generally takes the form: 1) represent the relational database as a graph, 2) relate the prediction task to a graph problem, 3) perform graph machine learning. Instead of using the SPADL data, we may see more success in using the more expressive Wyscout data [Pappalardo et al., 2019], which the SPADL dataset is derived from. The layout of these relational databases are visualized in the schemata 1 and 2. The learning task can be formulated as node classification: predicting the categorical label of the next action (referred to as an event in the Wyscout dataset) given a sequence of actions.

References

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018. URL https://arxiv.org/abs/1710.10903.

Joshua Robinson, Rishabh Ranjan, Weihua Hu, Kexin Huang, Jiaqi Han, Alejandro Dobles, Matthias Fey, Jan E. Lenssen, Yiwen Yuan, Zecheng Zhang, Xinwei He, and Jure Leskovec. Relbench: A benchmark for deep learning on relational databases, 2024. URL https://arxiv.org/abs/2407.20060.

Luca Pappalardo, Paolo Cintia, Alessio Rossi, Emanuele Massucco, Paolo Ferragina, Dino Pedreschi, and Fosca Giannotti. A public data set of spatio-temporal match events in soccer competitions. *Scientific Data*, 6(1), October 2019. ISSN 2052-4463. doi: 10.1038/s41597-019-0247-7. URL http://dx.doi.org/10.1038/s41597-019-0247-7.

A Appendix

Action Type	F1-Score
bad_touch	0.000
clearance	0.199
corner_crossed	0.000
corner_short	0.000
cross	0.055
dribble	0.000
foul	0.000
freekick_crossed	0.000
freekick_short	0.027
goalkick	0.008
interception	0.359
keeper_save	0.316
pass	0.813
shot	0.105
shot_freekick	0.000
shot_penalty	0.000
tackle	0.000
take_on	0.000
throw_in	0.156

Table 1: The best per-class F1 scores for our GAT model. Notice that there is a significant class imbalance.

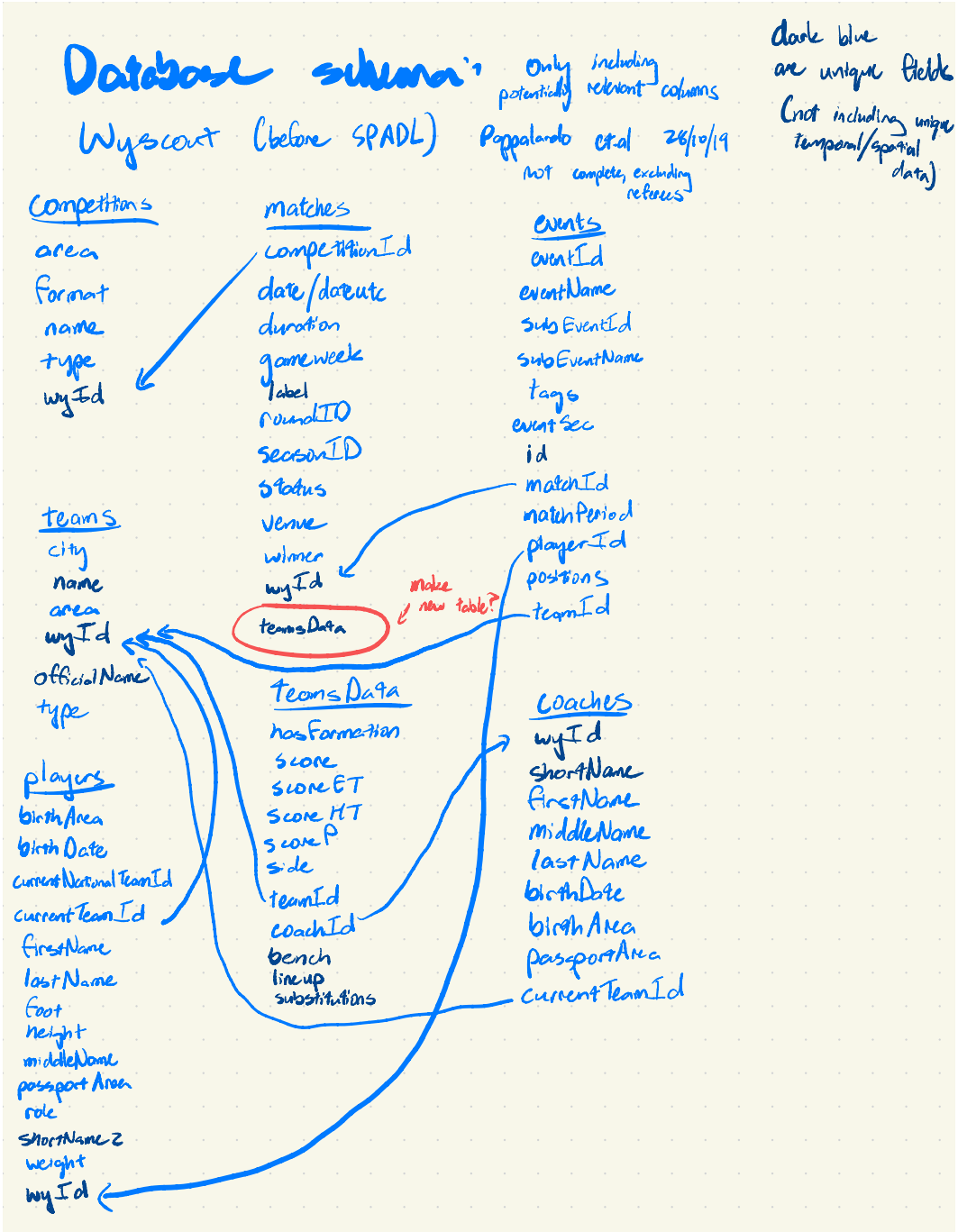

Figure 1: A database schema of the Wyscout dataset we use in the RDL model.

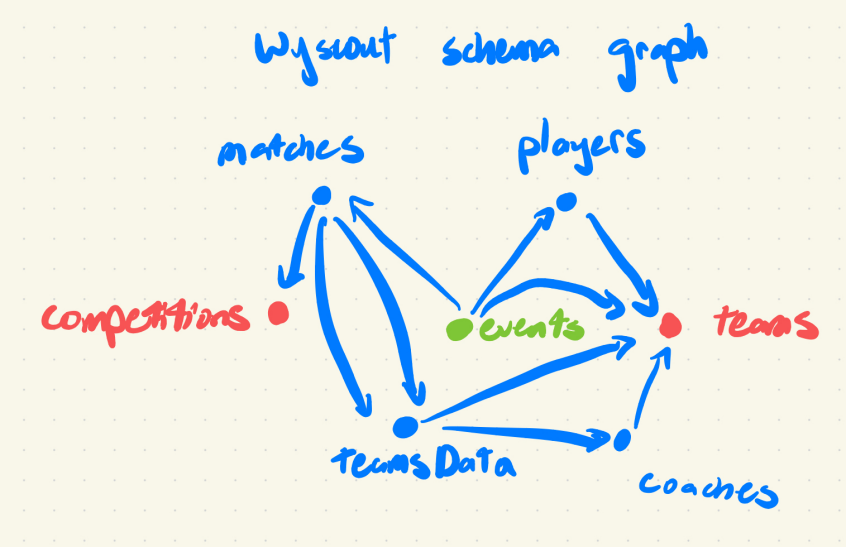

Figure 2: The above database schema translated into a graph, used for the RDL model.