

# Determining the phases of play using Graph Neural Network Embeddings

juan.campos@geniussports.com

Juan Camilo Campos

## Abstract

Determining the phases of play allows us to have a better understanding of a football game and to develop multiple applications such as characterizing the playing style of teams and players, or being able to search and analyze specific moments in a game. However, there is no consensus on how to define the different phases of a game; each team or football specialist has its own definitions, and labeling them can be a subjective task. Thus, instead of building a model that determines the phases of play, this work aims to create a tool that allows football specialists to build models that help them to identify those phases based on their own definitions. To achieve this goal, this work proposes a novel architecture for computing continuous vector representations of pass actions. These embeddings are generated using the information of the players' spatial distribution when the pass occurs using Graph Convolutional Networks, and capture football semantic properties.

---

## 1. Introduction

In football having the ball for longer periods during a game, does not imply that a team is more effective or superior. In fact, during La Liga 2013-2014, Atletico de Madrid showed a very effective playing style where, even though its opponents normally had a higher possession of the ball, Atletico ended up outperforming them and winning the championship. In order to systematically characterize unique playing styles, like the one that Atletico de Madrid showed during that season, we need to distinguish the phases of play in football games, which describe what teams do with the possession of the ball.

Maintenance, build-up, sustained-threat, and counter-attack are some examples of phases of play proposed by some football specialists [1]. However, there is not a broad consensus and the definition of the phases of play is highly dependent on the criteria of each specialist or football team. For instance, consider the definition of a counter-attack. In the football language, a

counter-attack is when a team regains possession and moves the ball into an attacking area with high speed before the opponent can recover their defensive shape. Note that both, *transition speed* and *defensive-shape recovery* are subjective elements in this definition. For this reason, instead of developing a model which automatically identifies certain phases of play, what if we design a tool that allows each person or football specialist to identify the phases of play based on their own definitions?

This work introduces a novel framework based on Graph Neural Networks to generate a vector representation of each football action; in particular, this work focuses on the passes, which are the most frequent football action. These vector representations —or embeddings— encapsulate information related to the context, such as, pass location, players position on the field when the pass occurs, or the actions that happen around it; and information related to the execution of the pass, such as, its final location or the body part used to perform it. Thus, based on these embedding properties, the football specialist would be asked to label a few examples from which a model that identifies a phase of play using the specialist’s own definition is built.

The remaining sections are organized as follows: section 2 introduces some previous works; section 3 presents the proposed framework to generate the football-action embeddings; section 4 describes the implementation of the proposed framework and a quality evaluation of these embeddings; section 5 builds a model to identify the maintenance phase based on the generated embeddings; and finally, section 6 draws some conclusions and future research directions.

---

## 2. Related Work

Embeddings are *low-dimensional-learned* continuous vector representations of discrete variables. They have been extensively used in the natural language processing field where, based on word embeddings, it is possible to leverage the performance of translators or auto-complete recommendation systems. The authors in [2] propose a technique called Word2vec, that uses neural networks to generate word embeddings. They propose two different neural network architectures; however, the idea behind both architectures is the same: to generate the word embeddings based on the context where the words normally appear in the usage of the language. It allows the embedding to capture some semantic properties.

In football, the use of embeddings for particular applications has started to come up in the last few years. In particular, the work in [3] uses the public StatsBomb dataset, along with some

open-source implementation of the Word2vec algorithm, to generate a vector representation of the football actions. Then, the author takes advantage of these embeddings to generate a vector representation of the players available in the dataset. This work is based on previous versions of the StastBomb data which do not include any information regarding the position of the players when a given event occurs.

During the past year, StatsBomb360 has taken the event data to the next level by incorporating the locations of the on-screen players in each event. This new data could give us the possibility to have a better understanding of the game and, at the same time, the need to face new challenges; for instance, to find the most suitable feature representation to exploit this highly variable and unstructured data.

Traditional representations tend to produce handcrafted features based on football knowledge. Even though this approach does not require high amounts of data to train the models, it can be computationally expensive to run them in real-time and can generate biased systems. To overcome these issues, the work in [4] introduces a novel approach where Graph Neural Networks (GNNs) learn directly from tracking data and produce real-time predictions for different football tasks. The authors show how GNNs can be applicable in order to model defensive behavior, however, this method is extendible to other football tasks and applications.

---

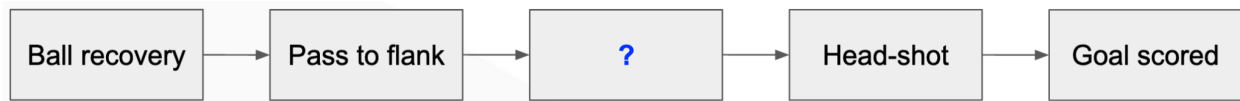
### **3. Proposed Framework - Pass2Vec**

In order to generate the vector representations —or embeddings— that summarize the most important information of each football action we must consider that:

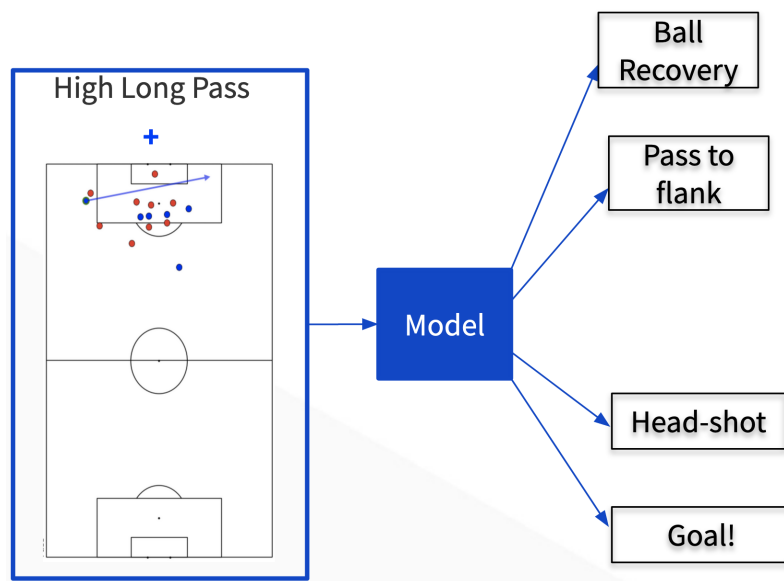
- Actions that occur in a similar context must be closely located in the space.
- Players' distribution on the field when an action occurs must be included.

Based on the above-mentioned bullets, we looked for a method to generate the embeddings in a novel way and discovered the concept of Word2vec. This concept is coined from natural language processing and generates the vector representation of a given word based on the context where the word normally appears. The authors in [2] propose two neural network architectures to produce the words' embeddings: CBOW, and Skip-gram. In CBOW, we predict a missing word located in the middle of a sentence through a model that uses known words surrounding the absent one as the input. Conversely, the Skip-gram architecture predicts missing words surrounding a known word located in the middle of a sentence that a model uses as the input.

One key assumption of Word2vec is that language is predictable and therefore, it is possible to design a model that infers missing information based on the context. As language, football is predictable, and we can try to predict a missing action based on the previous and following actions. For instance, consider the following sequence of actions:



Note that it is predictable that the missing action is a cross pass to the area. Consequently, inspired by the Skip-gram architecture, our approach proposes the architecture Pass2vec, to generate football action embeddings. As its name suggests, our framework focuses on passes, but it can be extended to other types of actions. Fig. 1 depicts the main idea behind Pass2vec. It consists of predicting, within a certain range, the actions that occur before and after a given pass by considering as input: first, general attributes of the given pass, such as its height, length, and angle; and second, the players' spatial distribution when the given pass occurs.

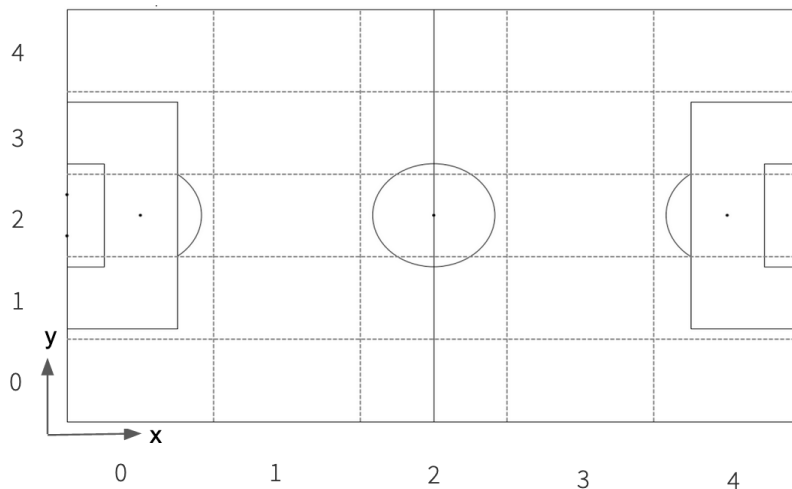


**Figure 1:** *Pass2vec architecture.*

Note that we only count with the position of the on-screen players for the Statsbomb360 dataset. This means that the number of players varies for every registered action. Then, we use a graph representation because it allows us to handle the freeze-frame data in a well-defined structure which does not require a fixed number of players in each action.

Let  $G(V, E, U)$  be a direct graph where  $V$ ,  $E$ , and  $U$ , denote the nodes, edges, and global attributes, respectively. Nodes represent players; its attributes are the player location, and two flags that indicate whether the player is the ball carrier or the goalkeeper. Edges represent the relationship between the players; its attributes are the distance, and the angle between them and a flag that indicates if both players are teammates. Ultimately, global features contain information regarding the execution of the pass; in particular, we include the location, height, length, angle, and the body part used to execute the pass.

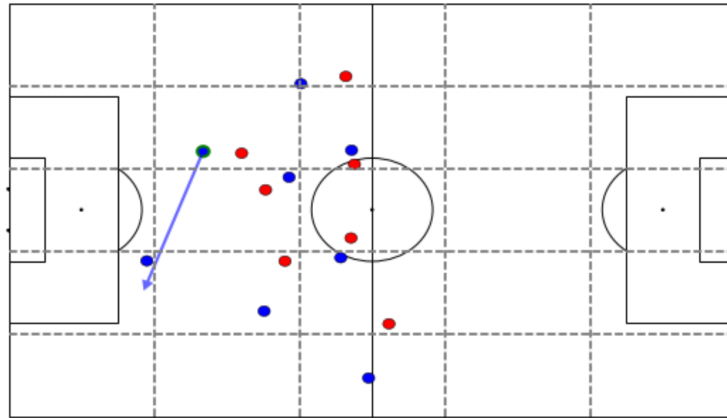
To represent the output of our model we convert each action into a word by following this structure[3]: “|Event name| |Event location| |Other attributes|”. For the |Event name| we consider all the events related to on-ball actions such as passes, shots, carries and dribbles. Actions like substitutions, half-starts, and tactical shifts are ignored. For the |Event location| we split the field into 25 rectangles to determine the location of the actions (see Fig. 2). And finally, for |Other attributes|, we consider additional arguments —depending on the action type— to generate the word representation of each event. For instance, if the action is a pass we consider the direction, the length, and the body part used to perform it; Fig. 3 depicts an example of the translation of a pass-action into a word.



**Figure 2:** Action location notation.

Now, we need to define the model architecture which receives the input graph and predicts the surrounding word-actions. For this purpose, we use a GNN followed by a weighted matrix. For the GNN, we apply the relation network structure which is an edge block followed by a global block [5]. Each block has a multilayer perceptron with three layers. The output vector from the global block represents our pass embedding. For the weighted matrix, we perform a dot product

between the pass embedding and each row of the weighted matrix, and to the resulting vector we apply a softmax activation function.



**Figure 3:** Ground pass, of medium length, executed with the right foot in the east direction.  
“|Pass| |1,3| |Ground-Medium-Foot-E|” is its word representation.

## 4. Experiments and Results

### 4.1 Data

We use the new StatsBomb360 event data of 540 matches of the Bundesliga and La Liga during the 2020/2021 season. Besides the traditional features included in this kind of data (such as the location, the type, or the outcome of the actions), this data provides the position of all on-screen players for almost every single event. There are 521037 passes and 98584 different ball possessions, where a ball possession represents a period of play where a single team is in control of the ball.

### 4.2 Training

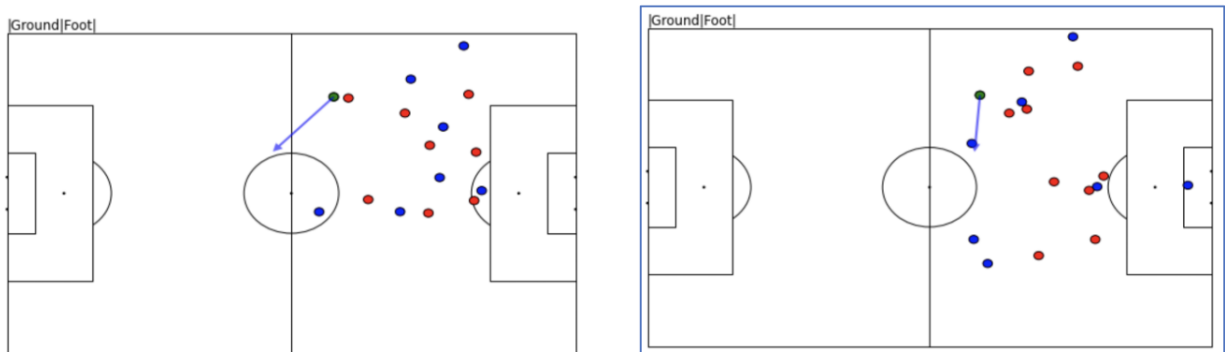
In Word2vec terms, ball possessions represent the sentences in a football match; that is, we iterate through each ball possession to determine the football actions that appear in the same context. Moreover, we use a negative sampling technique, which means that the objective in each training example will be to distinguish what the context word is, from a given set of negative examples. The negative sampling technique optimizes the training algorithm updating only a few values of the weighted matrix per each training sample. Thus, our training objective is to minimize the categorical cross-entropy loss using adaptive moment estimation (Adam). Table 1 lists the model hyperparameter values that we use to generate the embeddings.

Model hyperparameter	Value
Window size	3
Number of negative samples	5
Initial learning rate	$10^{-3}$
Batch size	1024
Number of epochs	15
Minimum frequency of word	5
Embedding size	32

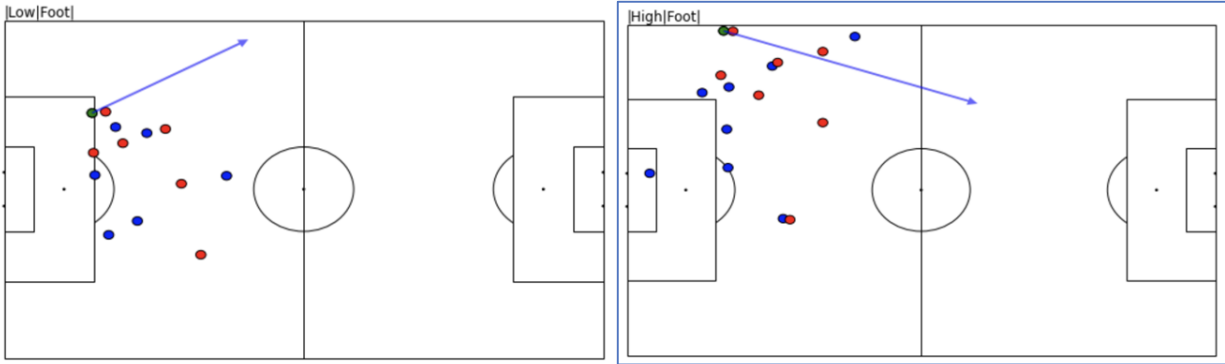
**Table 1.** Model hyperparameters

### 4.3 Results

To evaluate the quality of our embeddings we randomly choose a sample of passes from the data and find the most similar ones using cosine similarity. Figs. 4 and 5 show two examples where the pass on the left is randomly selected and the one on the right is its most similar, based on the embeddings. Note that in Fig. 4, even though each pass has a different end location, both happen in similar contexts: controlled ball possession and a defensive team well settled. Similarly, Fig. 5 shows two long passes which are also different in the execution (one is a low pass, and the other is a high pass), but for both, actions seem to be the beginning of a counter-attack.

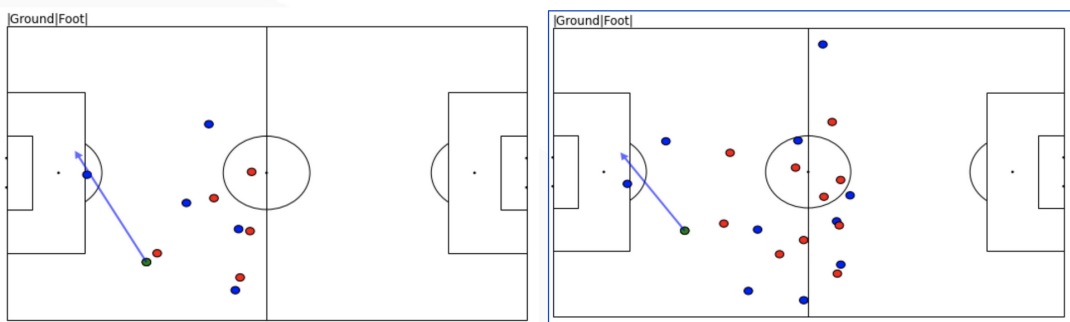


**Figure 4:** A backwards pass with its most similar embedding.

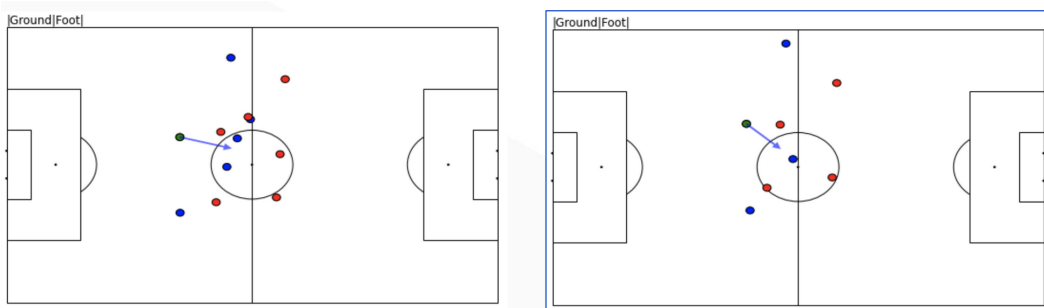


**Figure 5:** A long pass with its most similar embedding.

In Figs. 6 and 7 we intentionally select passes that must belong to a maintenance phase, and a build-up phase, respectively. Note that in both figures, the most similar pass seems to belong to the same phase of play. In Fig 6 both are passes to the goalkeeper, and the team in possession seems to keep the ball safe. Note, that our embeddings locate these passes closely together in the space, even though the number of players in each action is different. Similarly, in Fig. 7 both passes break the first line of defenders, and the team in possession seems to be in control of the situation.



**Figure 6:** Pass in maintenance phase with its most similar embedding.



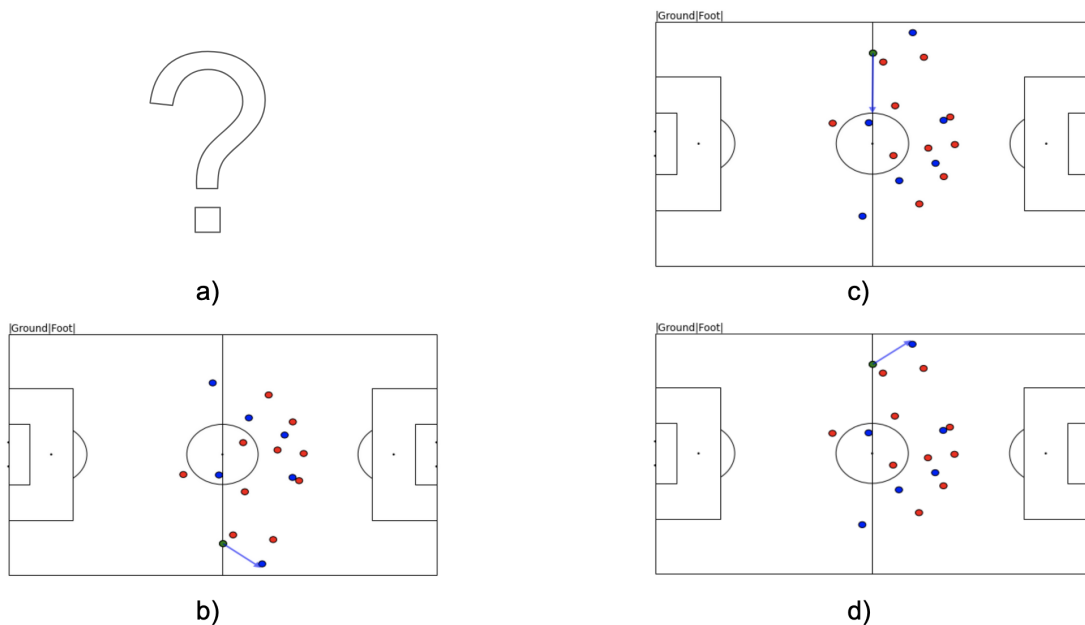
**Figure 7:** Pass in build-up phase with its most similar embedding.



Finally, inspired by the Word2vec evaluation method, we can use analogies to discover the football semantic properties that the embeddings are able to capture. InWord2vec, analogies aim to find relationships between pairs of words. Thus, the idea is to answer questions like “What is the word that is similar to *woman* in the same sense as *king* is similar to *man*?”[2] using the vector representations. Note that this question can be translated into vector operations as follows:

$$X = \text{vector}(\text{"woman"}) - \text{vector}(\text{"man"}) + \text{vector}(\text{"king"})$$

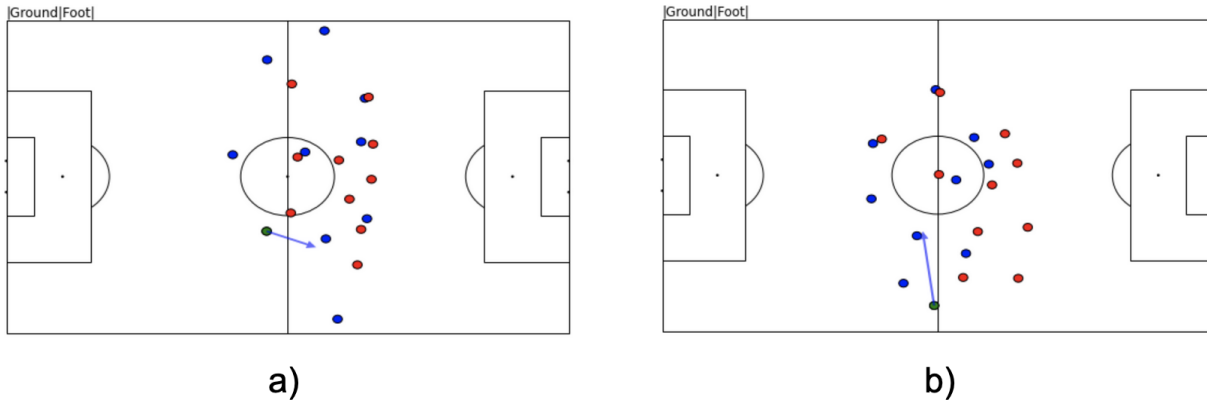
Unsurprisingly, the word “*queen*” has the most similar embedding to the resulting vector of the previous equation. Fig. 8 shows how we adapt this idea to the Pass2vec framework. Note that action c) and d) have the same players’ spatial distribution, the only difference is the direction of the pass. On the other hand, note that action b) is the mirror of action d), i.e., we flip the Y coordinates of the players location, and of the initial and final location of the pass. Then, we must expect that the solution to this analogy must be an action with a similar players’ spatial distribution of the action b) but the pass must start in the right flank and end in the center of the field.



**Figure 8.** Pass analogy: What is the pass a) that is similar to the pass b) in the same sense as pass c) is similar to pass d)?

Fig. 9a) shows the pass with the most similar embedding to the resulting vector of the previous analogy. Note that the final location of the pass is the center of the field, however, the initial

location is not in the right flank. On the other hand, Fig 9b) shows the pass with the second most similar embedding. Note that this pass is more similar to the expected pass, the pass starts at the right flank, ends in the middle of the field, and the players' space distribution is similar to the pass in Fig 8b).



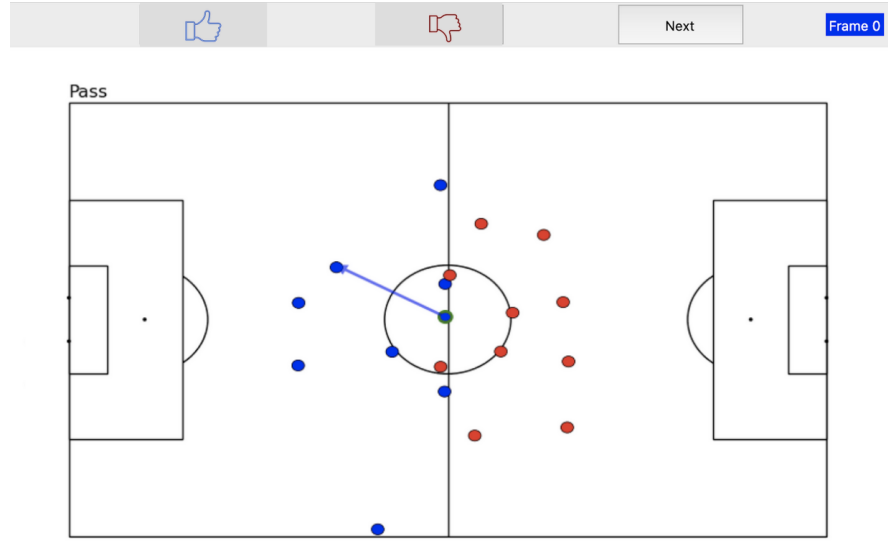
**Figure 9.** Passes with the most similar embeddings to the resulting vector of the analogy presented in Fig. 8

## 5. Practical Applications

Our final purpose is to make use of the generated pass embeddings to build a tool that allows football specialists to automatically detect the phase of play, based on their own definitions. Thus, we develop an UI (see Fig. 10) to systematically assign a tag to different passes in different matches. As an example, we focus on identifying the maintenance phase<sup>1</sup>. It means that for each given pass, we assign a label that defines whether or not the pass belongs to a maintenance phase, using our own definition of phases of play.

We tag around 500 different passes whose final location is the first half of the field (the own half of the team in possession). Then, we build a logistic regression model with these tagged passes as the training set. There are 343 maintenance passes, and 156 non-maintenance passes. The model achieves a training accuracy of 0.87.

<sup>1</sup> We define maintenance as the phase of play where the possession team looks to maintain and secure possession of the ball within its own half of the field [1].



**Figure 10:** The user interface for tagging passes.

Finally, using the trained model and a set of rules, we are able to identify moments of the game where there is a maintenance phase. In particular, we consider the sequence of actions where there are at least four consecutive passes executed by the same team, and all passes end in the first half of the field. Then, we classify a sequence of actions as maintenance if all its passes have a probability greater than 0.5 to belong to this phase ([link video](#)), otherwise, the sequence is classified as a non-maintenance ([link video](#)). This approach is useful to filter the moments of the game which we are interested in.

## 6. Discussion and Future work

The football actions embeddings give us the possibility to leverage the performance of models with different purposes in football, such as the identification of phases of play, or develop new applications, such as the search of similar actions. The proposed framework exploits the players' spatial distribution information, that the new StatsBomb360 data incorporates, to encapsulate football semantic properties in the vector representation of each action. This spatial data is highly variable and unstructured since it is obtained from the broadcast image, which means that the number of players varies for every registered action. We show how GNN is a proper technique to handle this type of data. We also present a practical application where, based on the generated embeddings, we build a model to identify the moments of the game when a maintenance phase occurs.

Adding new node attributes, such as the velocity and orientations of the players, or global attributes, such as the camera field-of-view, can improve the football semantic properties of the embeddings. Furthermore, developing a rigorous and systematic test to evaluate the quality of the embeddings remains as an interesting direction for future research.

---

## References

- [1] Stats Perform. (2020). Stats Perform Playing Styles – An Introduction. <https://www.statsperform.com/resource/stats-playing-styles-introduction/>.
- [2] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*
- [3] Magdaci, O. (2021). Embedding the Language of Football Using NLP. Towards Data Science. <https://towardsdatascience.com/embedding-the-language-of-football-using-nlp-e52dc153afa6>.
- [4] Stöckl, M., Seidl, T., Marley, D., & Power, P. (2021). Making Offensive Play Predictable-Using a Graph Convolutional Network to Understand Defensive Performance in Soccer. In *Proceedings of the 15th MIT Sloan Sports Analytics Conference*.
- [5] Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., ... & Pascanu, R. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.