Presenter: Ofir Ben Moshe Date: 19/11/23

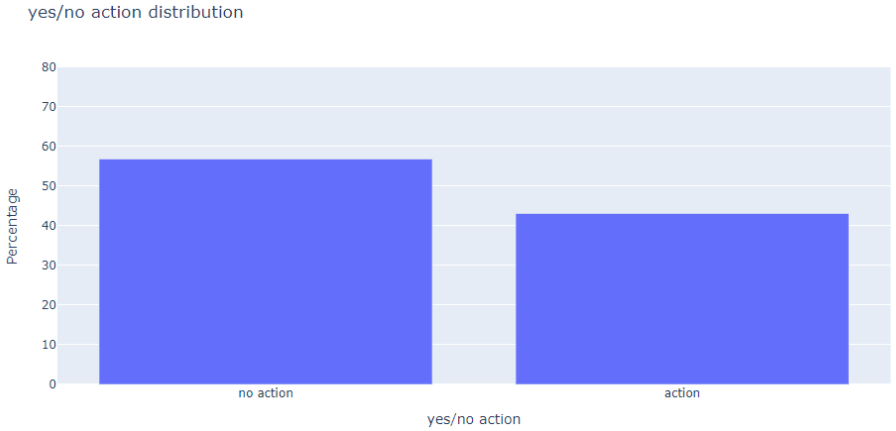
EDA- Action model

Information in terms of df size and the types of records:

```
There are 2592 records in the dataset
```

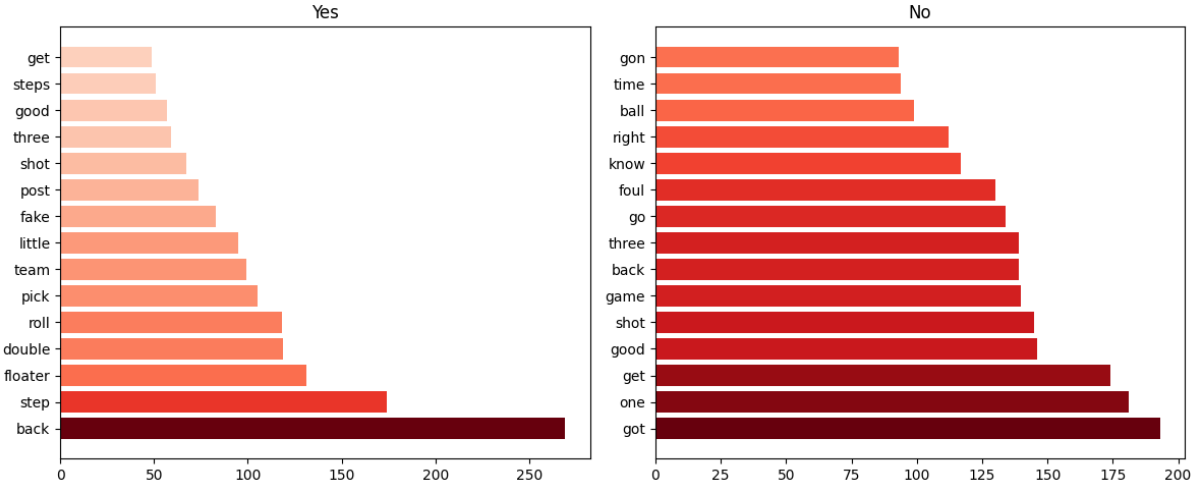
My assumption is that the model I will train will be able to distinguish what the action is when the action does exist, while it will be more difficult for him to decide in cases where there is no action.

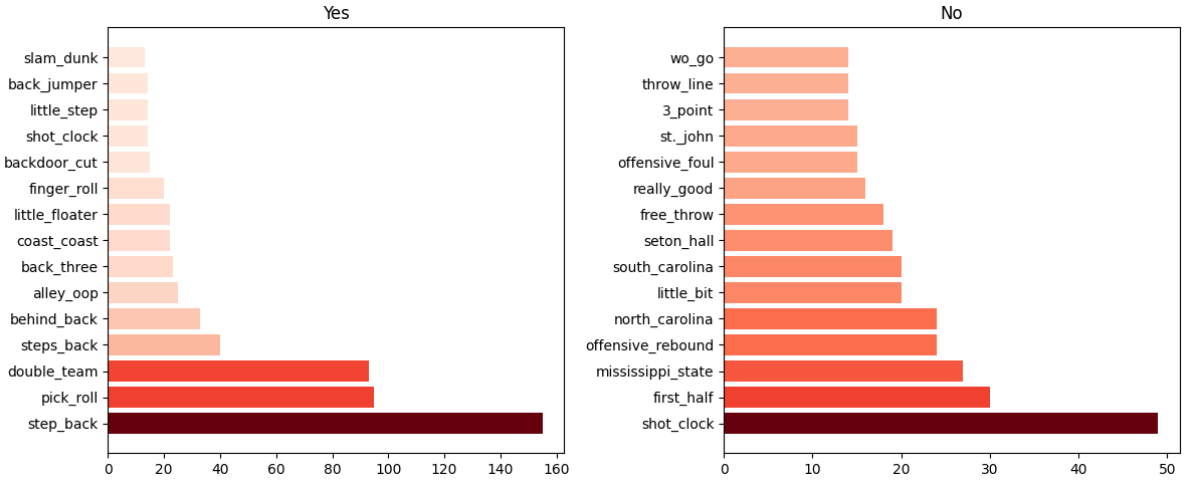
1. action and non-action records:



We can see that there are more records without action(57%).
From this analysis I understand that it is necessary to perform a preliminary step in which, given a transcript, it knows whether there is an action in the text or not.

I will now examine the distribution of words in relation to the two groups(without stopwords) - unigram and bigram:



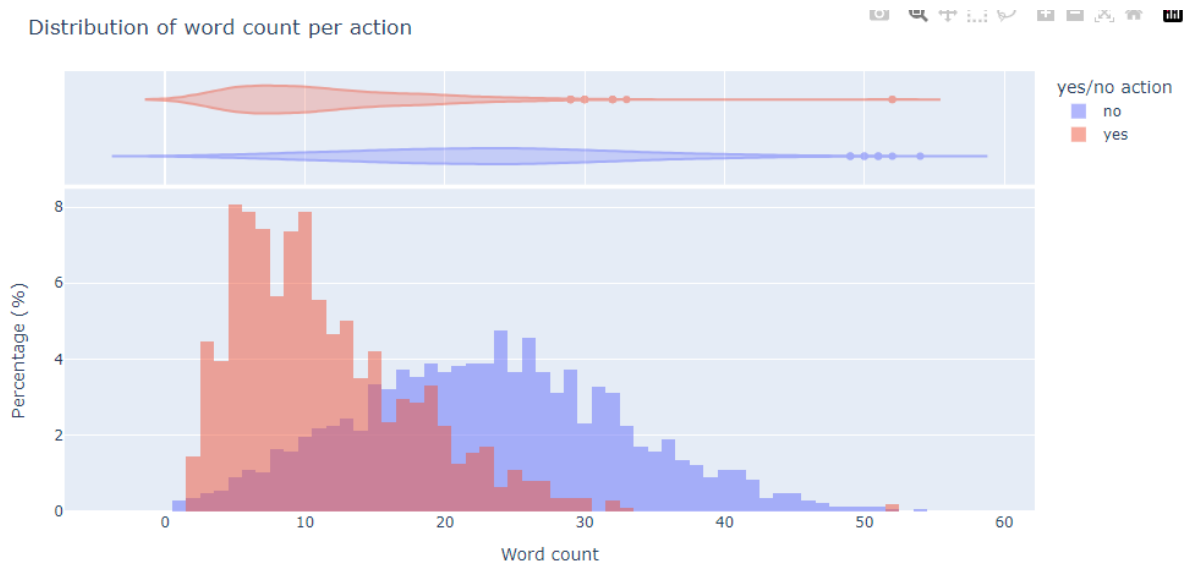


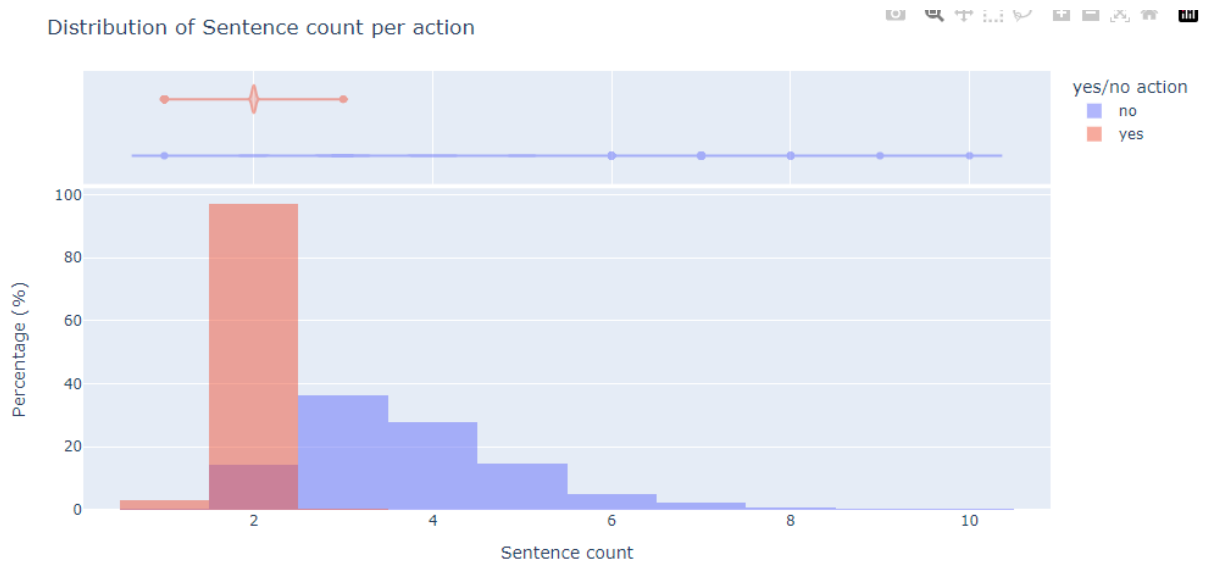
We can see the difference in the common words and also the difference in the distribution itself:

1. The most common words in texts in which there is an action are words which mostly refer to the actions themselves.
2. It can be seen that the frequency of the words is more exponential in the texts where action is present.

2. <u>Words/Sentences number distribution</u>:

One of the things I noticed is that transcripts without action tend to be longer:

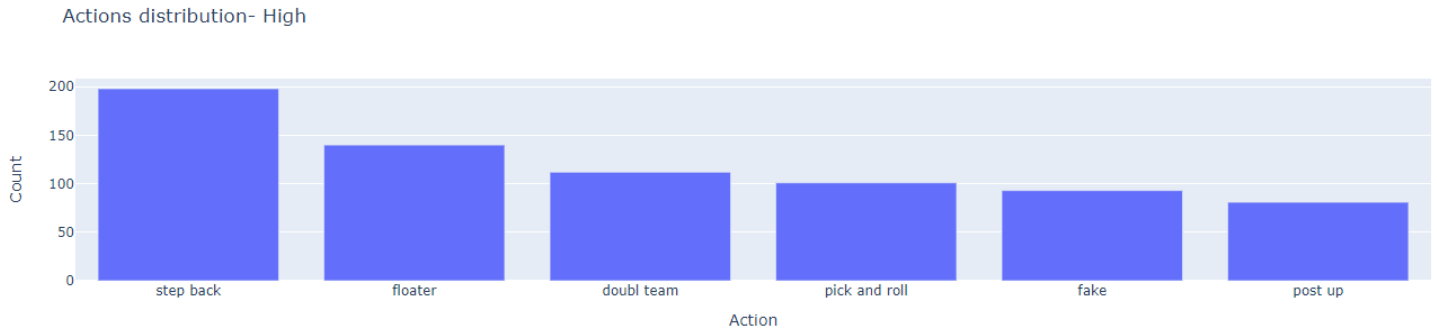


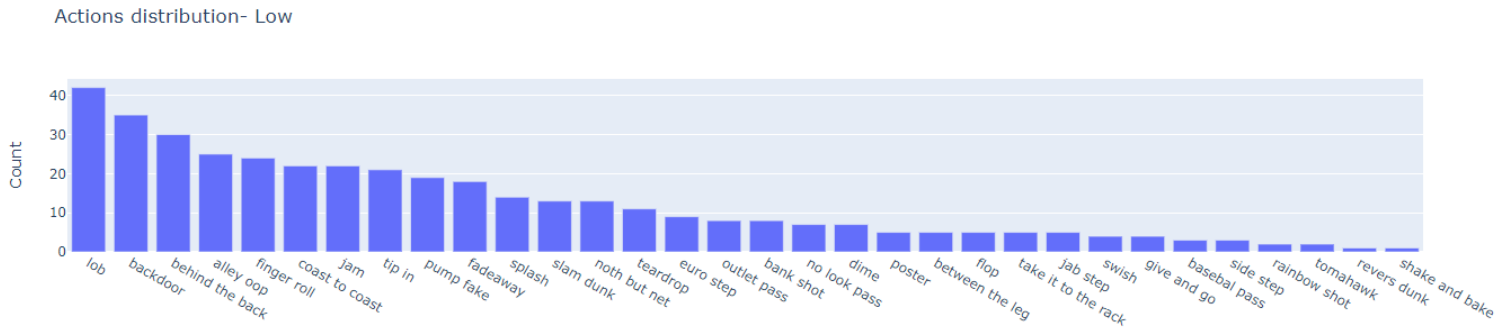


We can see the clear difference in the distribution of the number of words/sentences in the transcripts where there is no action and those that have action.

3. Actions distribution:

There are 38 actions(without 'No action')- a couple of them with high frequency, and all the rest with lower frequency, so I will present 2 graphs for better visualization(Common and less common actions).





In the first graph - action whose frequency is high, I assume this happens because these actions happen frequently during the game, therefore the broadcaster says them a lot.
It is important to refer to the fact that there are actions that exist only a few times in df(1 to 10), probably with a normal division into train and test, many of the uncommon actions will not appear in the test, therefore manual division is required.

Action Model:

My goal is to build a baseline model using Bert for action classification(including 'no action').

process:

1. split my DF into train and test for the purpose of training the model (80% for training).
   I decided that when a certain action has less than 2 records, all its records will be both in train and test.
2. Decision on parameters: batch size, learning rate, loss function and num epochs(I decided to reduce additional parameters such as Weight Decay, Learning Rate Scheduler, Warmup Steps and more due to time considerations).
3. Train BERT model and present results.

Optional Params:

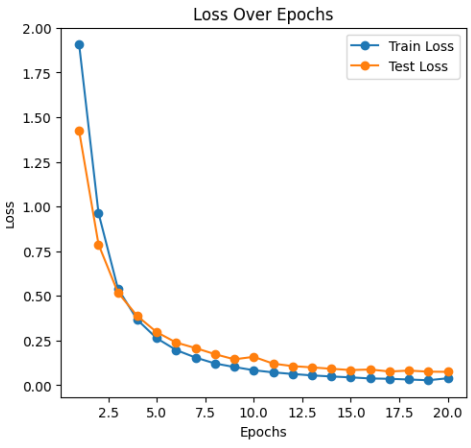
```
train_batch_size_lst = [8, 16, 32]
learning_rate_lst = [0.001, 0.0001, 2e-5]
criterion_lst = ['CrossEntropyLoss', 'MSE']
num_epochs = [5, 10, 15]
```

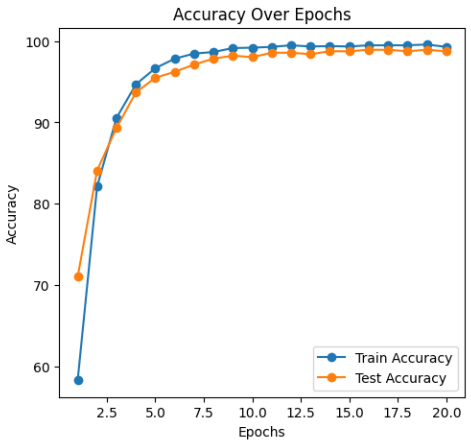
Best Params:

```
train_batch_size_lst = [16]
learning_rate_lst = [2e-5]
criterion_lst = ['MSE']
num_epochs = [10]
```

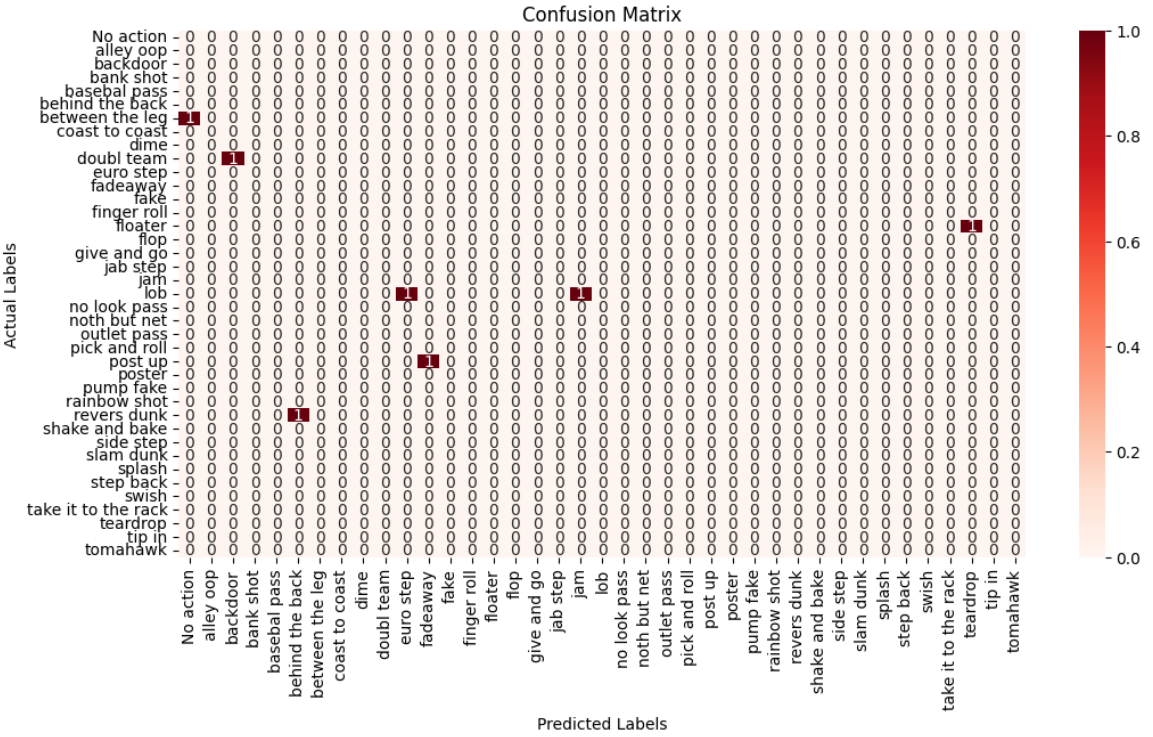
Results:

Average accuracy on 4 different random states on the Test set - 98.52%.
Average loss on 4 different random states on the Test set - 0.08.
Best model - Accuracy: 98.74%, Loss: 0.075.





I've dropped the values diagonally in the confusion matrix to **highlight** the mistakes.



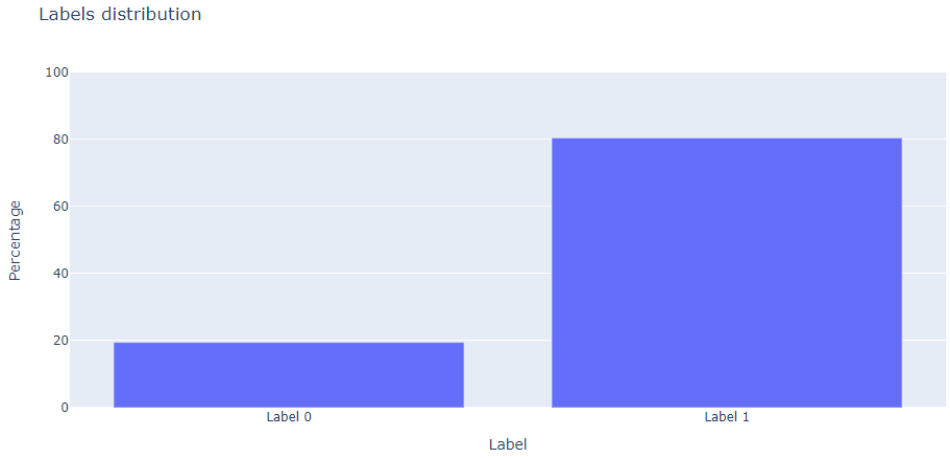
Drawbacks:

All in all, the model gives good results, so I don't see a fundamental flaw here. It is interesting to check which records the model falls on, because it is possible that the text contains several actions from the closed group, which can confuse the model in the classification of the correct action, but overall the error is marginal.

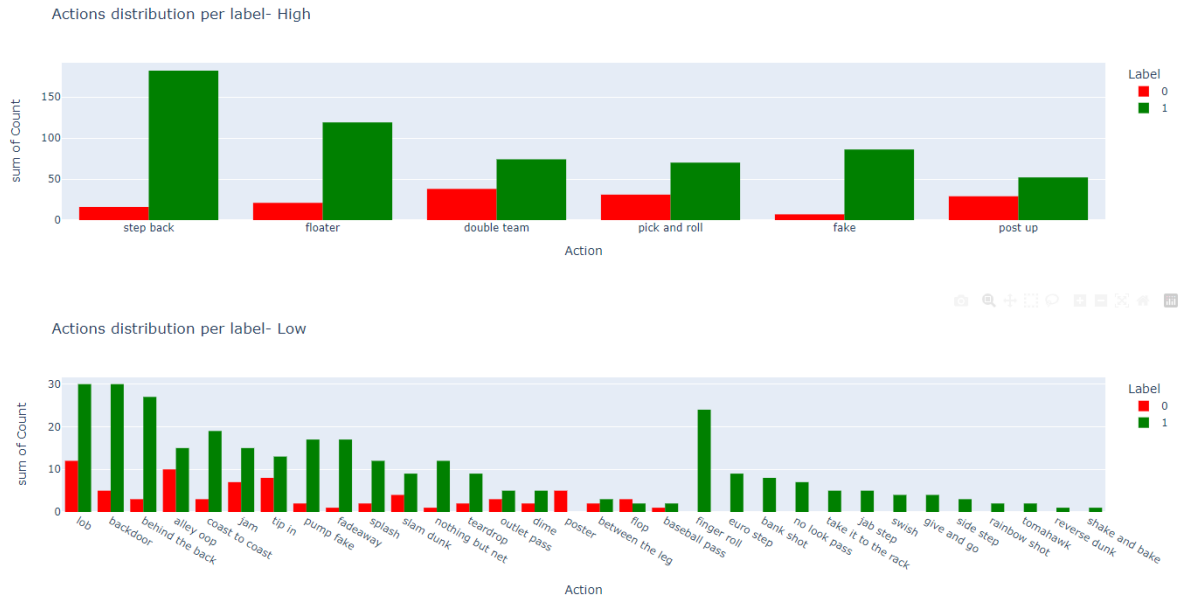
EDA- Validity model

The next step, after the action prediction step, is to predict whether the action is indeed valid in relation to the text.

First I conclude that when the model predicts that there is no action for the text, then the label is automatically 0, therefore I removed from the df all the records where the action is None.



We can see that here the situation is more complex - the df is not balanced and in most of the records the label is 1, meaning the action is valid for the text.



Here too, I separated the most common actions from the least common - and for each action I checked the number of valid/invalid records in the df.
And too, I continued with the previous approach of building the train and the test set.

My next step is to understand the <u>difference</u> between records that are valid and records that are invalid.

According to the definition listed in the task:

The action is considered valid if the action (e.g., "a floater shot") it describes is aligned with the action on the court.
An example of an invalid action could be a commentary description of an action that happened in the past or in previous events.

From the following definition I assume several things:

1. <u>An action that is valid will have a certain pos tag that is more common.</u>:
For example: ‘Jordan trying to post_up against Oxymora.’
Here the action is post-up and you can see that it is indeed the **verb**.

```
('trying', 'VERB'),
('to', 'PART'),
('post_up', 'VERB'),
('against', 'ADP'),
```

Another example: They're going to come and help on all of his post_ups.
Here the action is also post-up but you can see that it is **not the verb**.

```
('all', 'PRON'),
('of', 'ADP'),
('his', 'PRON'),
('post_ups', 'NOUN'),
('.', 'PUNCT')]
```

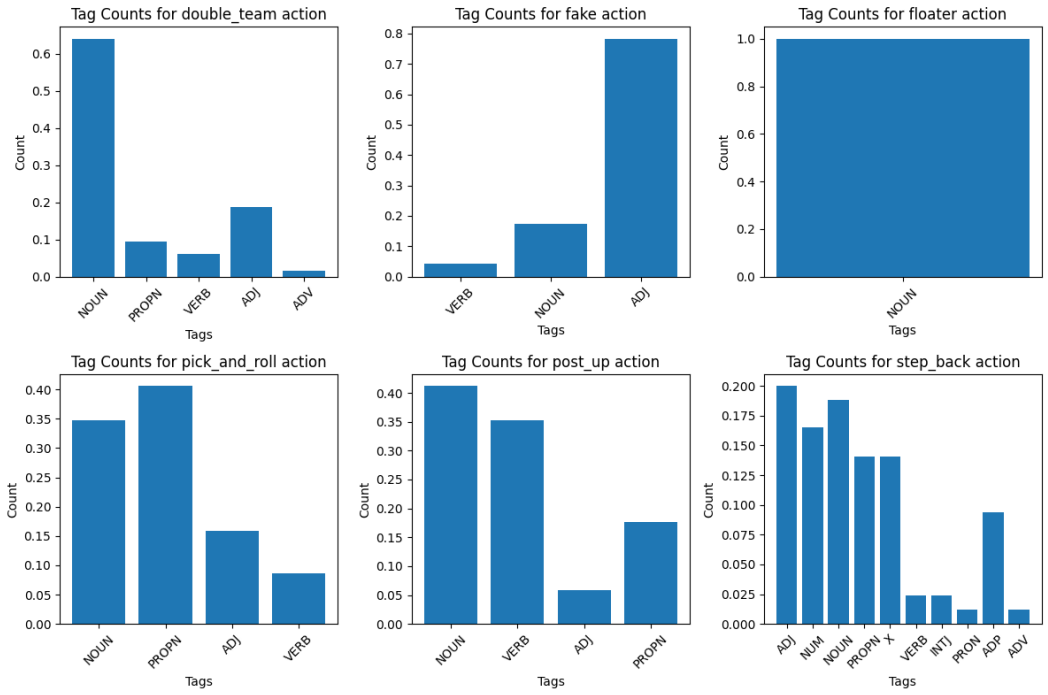
2. In text where the action is valid, the text will represent what is happening in the present:
For example: ‘Here comes Donovan Mitchell into the Lane double-teamed.’
The text refers to the player performing the double-team action in the present.

Another example: 'Well, I saw a different type of defense against the Raptors double-teamed every time he touched the ball is Westbrook runs is for the midrange shot of fibrosing.'
It can be seen that the broadcaster says that he 'saw' in past tense the double-teamed action.

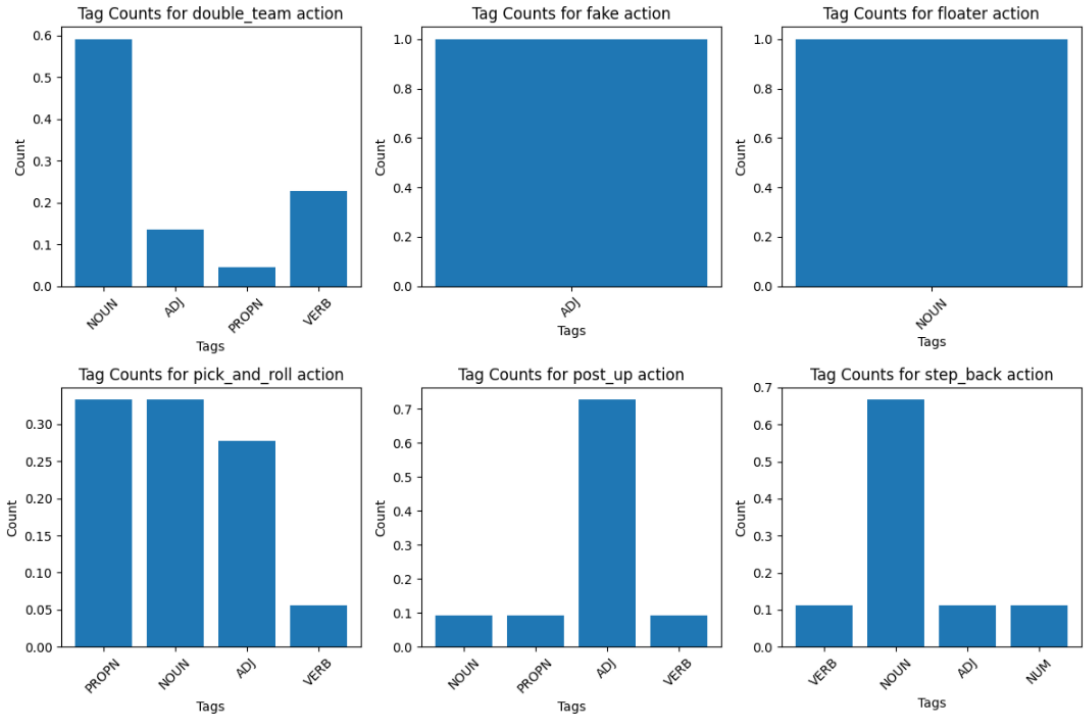
Graphs(All graphs are based on the train set)**:**

1. An action that is valid will often be used as a verb in the text(using spacy model):

Label is Valid



Label is Invalid

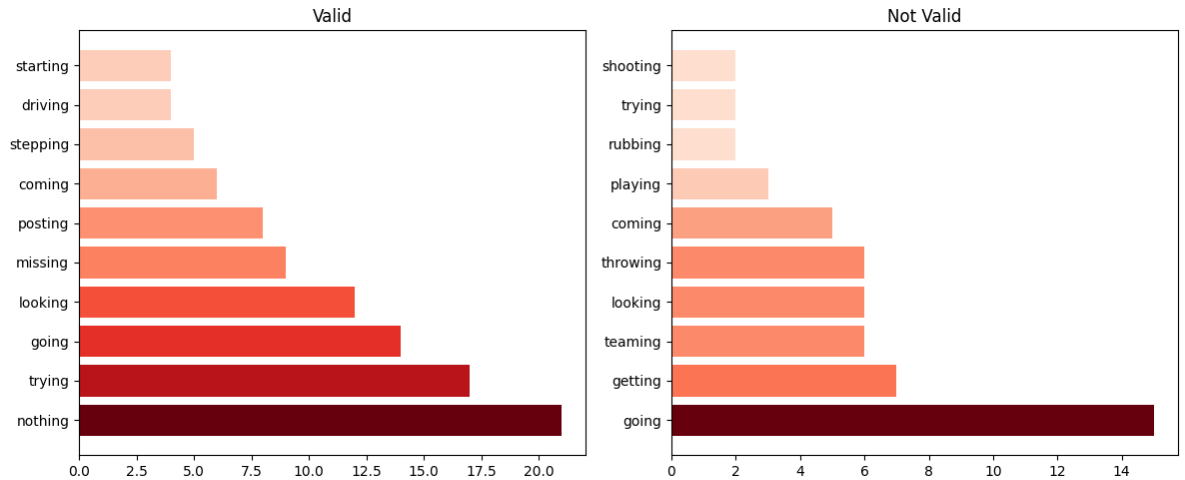


I chose the six most common actions to show the differences.
It can be noticed that in the texts where the action is valid, the pos tag distribution is more diverse.
Although, despite this, it is difficult to draw decisive conclusions regarding the difference between the distribution of the pos tags in the various labels.

2. When action that is valid, the text will represent what is happening in the present:

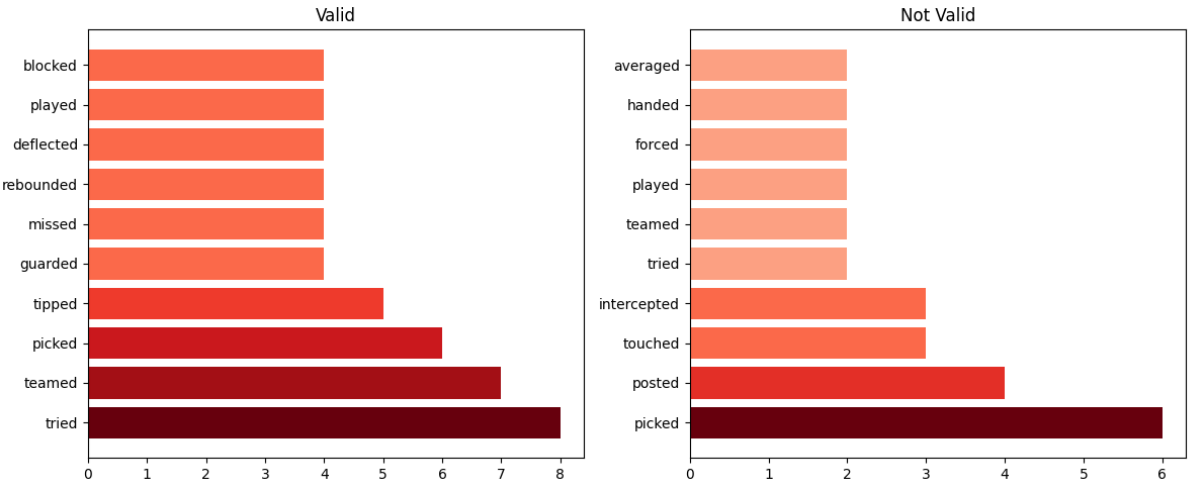
To examine the time of the text I decided to look for words whose endings are related to time, for example: ing, ed and ies.

ing



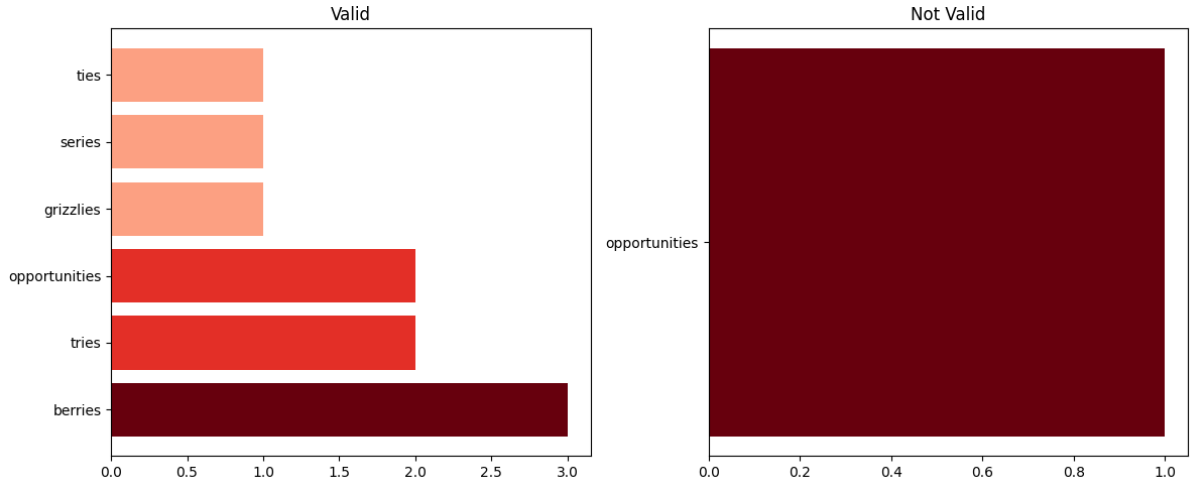
We can see that a word like nothing is very common in texts where the action is valid, and on the other hand a word like getting is common in texts where the action is invalid.

ed



We can see that a word like tipped is very common in texts where the action is valid, and on the other hand a word like posted is common in texts where the action is invalid.

ies



Here the situation is even clearer, you can see that except for one text in invalid, all the other texts in it do not contain words with the suffix ies.

For all these graphs it is also possible to perform an analysis at a **higher level of granularity**, for each action checking the distribution of words between the different labels.

Validity Model(Baseline):

My goal is to build a baseline model using Bert for validity binary classification. Because the database is unbalanced, I will base my decisions based on metrics that weight the accuracy between the two classes (f1 score, weighted accuracy) and based on the loss of the model.

Optional Params:

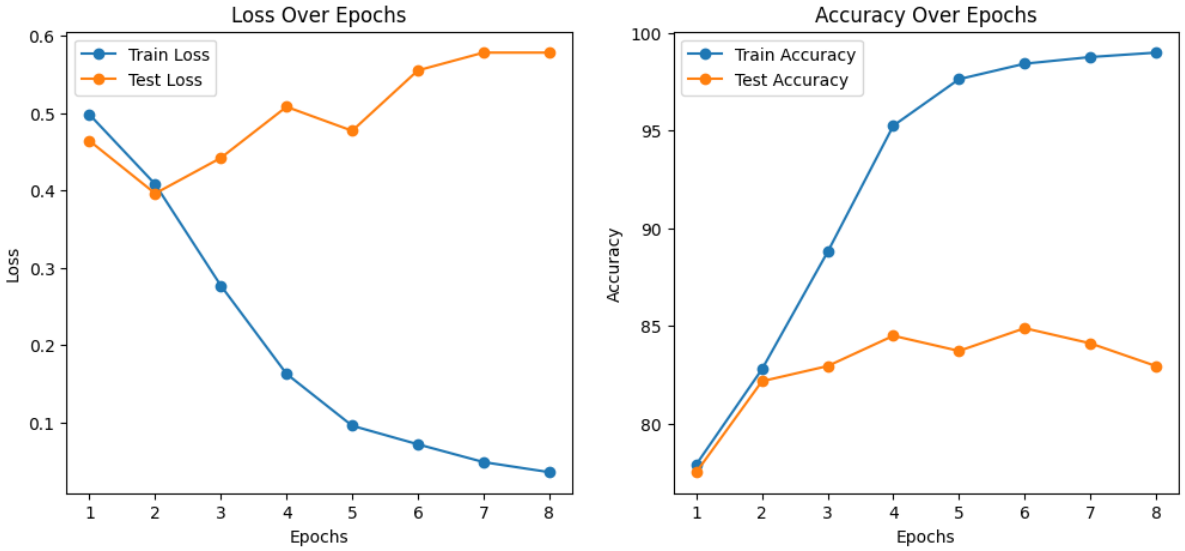
```
train_batch_size_lst = [8, 16, 32]
learning_rate_lst = [0.001, 0.0001, 2e-5]
criterion_lst = ['CrossEntropyLoss', 'MSE']
num_epochs_lst = [5, 8, 10]
```

Best Params:

```
train_batch_size_lst = [16]
learning_rate_lst = [2e-5]
criterion_lst = ['CrossEntropyLoss']
num_epochs_lst = [8]
```

Results:

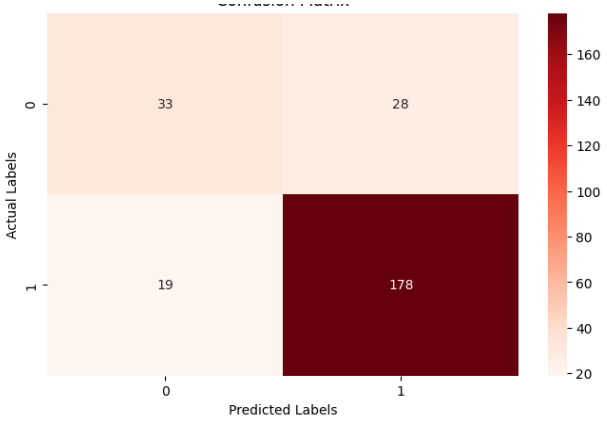
Average weighted accuracy on 5 different random states of the Test set - 80.18%.
Average loss on 5 different random states of the Test set - 0.42.
Best model - weighted Accuracy: 81.32%, Loss: 0.35.



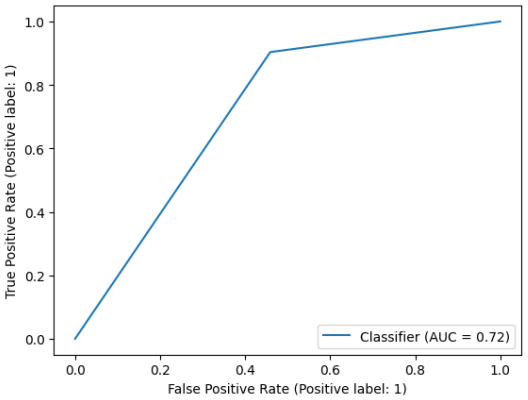
It can be seen that after several epochs the loss of the model does not improve on the test set and even starts to decrease, for this I used **early stopping** to keep the model with the lowest loss on the test set, assuming that the lower the loss on the test set, the better it generalizes.

Threshold tuning - I tried to change the threshold point of the model to improve its performance, based on f1 score.
optimum threshold : **0.354**

Confusion Matrix:



ROC-Auc graph:



Classification report:

	precision	recall	f1-score	support
Label 0	0.63	0.54	0.58	61
Label 1	0.86	0.90	0.88	197
accuracy			0.82	258
macro avg	0.75	0.72	0.73	258
weighted avg	0.81	0.82	0.81	258

Drawbacks:
The main problem with the current model is that it **"doesn't know"** what it is training for.
In the action model in the previous step, inside the text was the action itself, which made it easier for the model to classify the different actions.

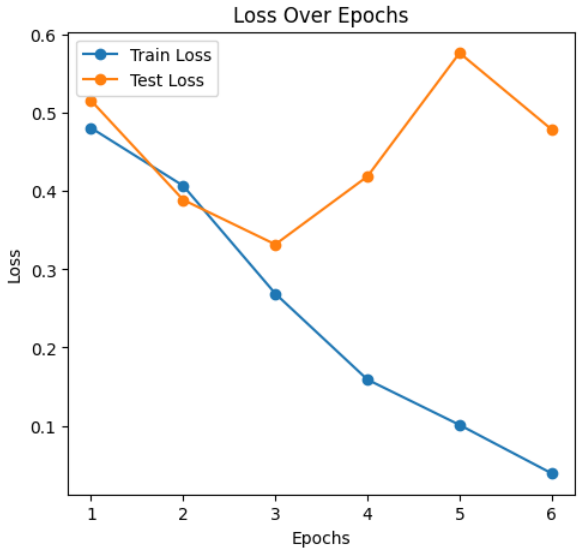
Here it is difficult for the model to understand exactly what he is training for, and what he is looking for.
To try to direct it more, it is useful to integrate in a certain way the action predicted in the previous model, so that the model understands what to focus the learning on.

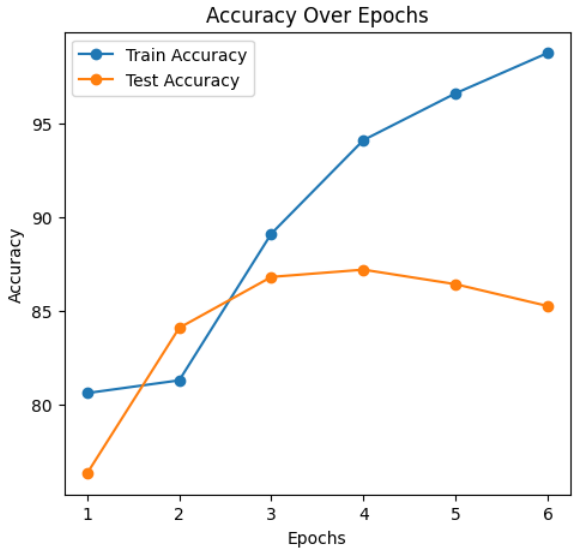
To combine the action with the text, I will use the **special character [SEP]**.
In fact, the action will be a kind of context for the text itself and may help it focus the learning.

```
Text + '[SEP]' + action
```

Validity Model(Improve):

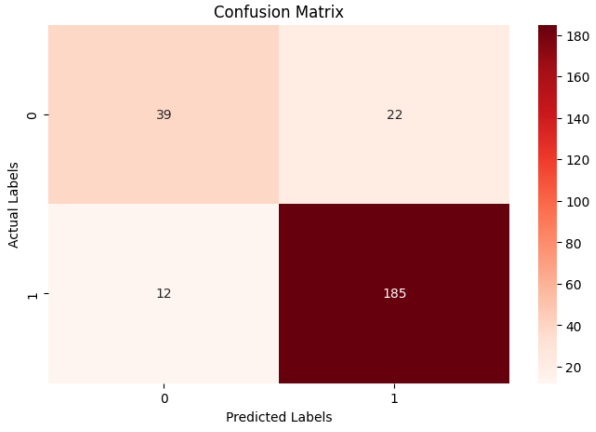
Average accuracy on 5 different random states of the Test set - 84.64%.
Average loss on 5 different random states of the Test set - 0.39.
Best model - weighted Accuracy: 86.42%, Loss: 0.33.



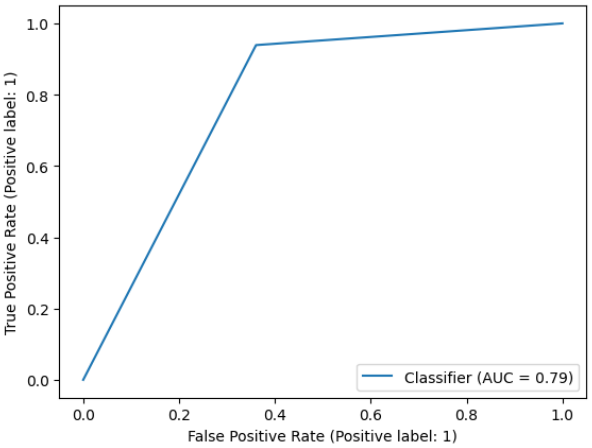


optimum threshold : **0.675**

Confusion Matrix:



ROC-Auc graph:



Classification report:

	precision	recall	f1-score	support
Label 0	0.76	0.64	0.70	61
Label 1	0.89	0.94	0.92	197
accuracy			0.87	258
macro avg	0.83	0.79	0.81	258
weighted avg	0.86	0.87	0.86	258

Conclusions:

1. The improved model brings better results (not significant) - in the improved model the loss is also lower (by 0.03), the weighted accuracy is higher (by 5%) and the AUC is higher (by 0.07).
2. The use of the optimum threshold helps to improve the accuracy of both models.

Final model

The final model is actually a pipeline of the last two models.
Given text, the pipeline first predicts the action. If it exists, it also predicts whether the action is valid or not.

Examples:

1

Given the **text**: "The other night Nurkic gets inside bank shot won't go the second attempt.".

the **Output** is: “The action is 'bank shot'.”

2

Given the **text**: 'Another solid possession, Richmond stepping up. Making the right play after turning it over comes back swing pass for kale. Rice now had that one altered.'

the **Output** is: “There is no action in the transcript.”

3

Given the **text**: "So he's going to use those shot fakes.".

the **Output** is: “The action 'fake' is not valid.”

Both models trained on the same train set.

Accuracy on test set: 86.076%

Model mistakes:

FP mistakes(real label is 0, predicted label is 1, on train set):

Text	Action
He can stretch you out, but also he is one of the best and being active around the glass and in the pick and roll.	pick_and_roll
He can stretch you out, but also he is one of the best and being active around the glass and in the pick and roll.	pick_and_roll
Alley Oop intercepted by Embiid Hornets.	alley_oop
Pat and step back.	step_back

There are a few cases here that, at least in my eyes, are quite unclear,
For example, the text: 'Alley Oop intercepted by Embiid Hornets.'
From how I understand it, it does refer to an action that corresponds to what is happening on the field, therefore if it is really a mistake it is very difficult to understand it by the model.
Another example: 'Pat and step back.' Also, very unclear.

FN mistakes(real label is 1, predicted label is 0, on train set):

Text	Action
At his best, that's when he can be super effective when you can pick and roll to the basket.	pick_and_roll
At his best, that's when he can be super effective when you can pick and roll to the basket.	pick_and_roll
But when he comes cross selling off their brush green, he's a good post up player.	post_up
If you go into that defensive circle and post up, you notice the defensive players behind you and policy and left that jump look a little bit short.	post_up
And also Westbrook guys that can defend post up players in this League.	post_up
We got double team.	double_team

Also here there are a few cases that unclear,
For example, the text: 'But when he comes cross selling off their brush green, he's a good post up player.'
From how I understand it, the text refers to the player being a good post-up player, and not to the action that takes place on the court.
On the other hand, for the text: 'We got double team'.
It seems that the model was wrong, maybe because of the word 'got' which refers to the past tense.

<u>paths:</u>
Datasets/action_enrichment_df_train.csv
Datasets/action_enrichment_df_test.csv
Models/bert_action_model.pth
Models/bert_validity_model.pth