# **Documentation**

This is the documentation of the event data transformation from Instat (in xml format) to Wyscout (in json format). The raw xml file is first transformed into a pandas dataframe and processed there afterwards. After the processing the dataframe is converted into the correct json format.

The code was written in Python 3.10.4 using Pandas 2.1.1 and NumPy 1.26.2.

All referred files and documents can be found under: https://github.com/Rui1818/Instat\_to\_wyscout\_data\_transform

## Preprocessing

Before the actual event data, Instat includes a lot of information that is irrelevant for the Wyscout format (like average player position and substitute player). So, we drop the irrelevant rows first after we get the pandas dataframe. But before that the Keepers and Formations get extracted and saved in global arrays together with the according teams (KeeperA,KeeperA for the Keepers, and teamA, teamB for the formations). For example KeeperA can look like this: [M.Neuer, Bayern München] (first the name then the team, same for the formations). At the end, the possession attributes for the first possession are set for the transformation. After preprocessing the first row/event is the start of the game (kick off).

## Transformation

The core of the transformation is the create\_event function. For each invocation it creates at least one Wyscout event and completes the entries of the Wyscout attributes. A single Wyscout event/entry usually contains multiple Instat entries. Each Wyscout event contains a primary tag (like pass, shot, duel etc.) with multiple secondary tags (in form of a list) that describes the primary. For that reason, we introduce *action triggers* here. Each *action trigger* is an entry of the Instat ‘action\_name’ attribute. Each time we encounter an *action trigger* a new Wyscout event is created, and we go through the Instat entries until the encounter of the next *action trigger* to fill up the list of secondary tags. A list of all *action triggers* called primary\_events can be found at the top of the transformations.py file.

At some occasions the create\_event function also creates a second linked event besides the one given by the *action trigger.* This is the case when:

* The first event contains a *shot on target/goal* -> creating an event with a *shot\_against* primary
* The first event contains a *duel* ->creating an event with a *duel* primary in perspective of the opponent
* The first event contains *ball\_out* (ball is played out of the field) and there is no *goal*\_kick/throw\_in afterwards-> creating an event with a *goal\_kick* resp. *throw\_in* primary depending on where the ball went out. Here there is also a *generated* tag in the secondary to differentiate which action was given by the data and which was self generated.

All entries (except names) are filled out in the correct Wyscout formats. For more information about the format refer to the Wyscout documentation: <https://apidocs.wyscout.com/>.

At the end in the pandas\_to\_json.py file the processed dataframe gets formatted and and transformed in a Wyscout compatible json format.

## Other remarks

There was no documentation for the Instat format. For that reason, the lists of action names might be incomplete. All actions were extracted from a dataset with 30 games from all age categories. A list of all found Instat actions and positions can be found in the ‘Instat\_information’ folder in the Git repository.

Also player/team names might be slightly different due to the different naming style for each company.

In general, most transformed attributes fits pretty well to the reference data only with some slight differences due to inconsistencies. There were also attributes with bigger differences: all the ‘touch’ events, because there is no exact transformation from the Instat data. There is a code snippet in the instat\_to\_wyscout\_pandas.py file, where for each time a player moves at least 5 meters with the ball, we create a touch event. This part was taken out (commented out), because this was too speculative. Also, the line between interceptions and duels were not so clear between both data sources. An action can be classified as an Interception for Instat but a duel for Wyscout and the other way around, so there is a certain overlap between the 2 attributes.

# Event data structure

This part lists all Instat and Wyscout attributes. For the Wyscout attributes there is also a explanation how the attributes are derived from the Instat structure.

## Instat event structure

These are the attributes from the raw Instat event data. Not all attributes are used for the transformation. Some attributes do not always appear, they are still mentioned below but not used in the final transformation.

|  |
| --- |
| Attributes |
| id |
| uid |
| number |
| action\_id |
| action\_name |
| player\_id |
| player\_name |
| team\_id |
| team\_name |
| half |
| second |
| standart\_name |
| ts |
| dl |
| position\_id |
| position\_name |
| opponent\_id |
| opponent\_name |
| opponent\_team\_id |
| opponent\_team\_name |
| opponent\_position\_id |
| opponent\_position\_name |
| zone\_id |
| zone\_name |
| zone\_dest\_id |
| zone\_dest\_name |
| pos\_x005f\_x |
| pos\_y |
| pos\_dest\_x005f\_x |
| pos\_dest\_y |
| len |
| possession\_id |
| possession\_name |
| possession\_team\_id |
| possession\_team\_name |
| possession\_time |
| possession\_number |
| direction |
| attack\_status\_id |
| attack\_status\_name |
| attack\_type\_id |
| attack\_type\_name |
| attack\_flang\_id |
| attack\_flang\_name |
| attack\_team\_id |
| attack\_team\_name |
| attack\_number |
| gate\_x005f\_x |
| gate\_y |
| body\_id |
| body\_name |
| shot\_type |
| touches |
| shot\_handling |
| assistant\_id |
| assistant\_name |

## Wyscout event structure

Words between ‘’ are attribute keywords or lists of entries of ‘action\_name’. Instat and Wyscout attributes can be differentiated by checking the strings, Wyscout separates words by a dot and Instat by a underline. For the *duel* primary the attributes are set for either ‘aerialDuel.’ or ‘groundDuel.’ depending on the tag in the secondary with the same name, the other is always set to unknown.

All lists for the tags can be found on the top of the transformations.py file.

‘possession.’ attributes are set a bit differently. They stay the same for multiple events. Once the Instat ‘possession\_duration’ attribute is >0 (signaling a new possession) a new possession is set.

|  |  |
| --- | --- |
| Attributes | description |
| id | Is set to ‘id’ (not the first appearance of it), not set to 0 for potential future usage |
| matchId | Is set to the first appearance of ‘id’, not set to 0 for potential future usage |
| matchPeriod | Transformed to the correct format from ‘half’ |
| minute | Transformed from ‘second’ |
| second | Transformed from ‘second’ |
| matchTimestamp | Transformed from ‘second’ (Offset is set to 0 for now) |
| videoTimestamp | Transformed from ‘second’ |
| relatedEventId | Is set to 0 due to incompatibility between Instat and Wyscout Id’s |
| type.primary | See tag documentation |
| type.secondary | See tag documentation |
| location.x | Transformed from ‘pos\_x005F\_x’ |
| location.y | Transformed from ‘pos\_y’ |
| team.id | Is set to 0 due to incompatibility between Instat and Wyscout Id’s |
| team.name | Is set to ‘team\_name’ |
| team.formation | Formations are extracted in preprocessing, more in the preprocessing section |
| opponentTeam.id | Is set to 0 due to incompatibility between Instat and Wyscout Id’s |
| opponentTeam.name | Can be extracted from the preprocessing, more in the preprocessing section |
| opponentTeam.formation | Formations are extracted in preprocessing, more in the preprocessing section |
| player.id | Is set to 0 due to incompatibility between Instat and Wyscout Id’s |
| player.name | Is set to ‘player\_name’ |
| player.position | Transformed from ‘position\_name’ |
| pass.accurate | Lookup ‘action\_name’ in the ‘accurate\_pass’ list, set to true if ‘action\_name’ value is in the list else false |
| pass.angle | Calculated passing angle in Wyscout format using the origin resp. destination coordinates from ‘pos\_x005F\_x’, ‘pos\_y’, 'pos\_dest\_x005F\_x', 'pos\_dest\_y' |
| pass.height | Is set to 0 due to missing attribute |
| pass.length | Is set to ‘len’ |
| pass.recipient.id | Is set to 0 due to incompatibility between Instat and Wyscout Id’s |
| pass.recipient.name | Calculated by extracting ‘player\_name’ from the next future Instat event with a player different than the current ‘player\_name’ and the same ‘team.name’, if the current ‘pass.accurate’ is true, else set to unknown |
| pass.recipient.position | Same procedure as for ‘pass.recipient.name’ but extracting the transformed ‘position\_name’ |
| pass.endLocation.x | If ‘type.primary’ is not shot, set to 'pos\_dest\_x005F\_x' else to unknown |
| pass.endLocation.y | If ‘type.primary’ is not shot, set to 'pos\_dest\_y' else to unknown |
| possession.id | Is set to 0 due to incompatibility between Instat and Wyscout Id’s |
| possession.duration | Is extracted from ‘possession\_time’ for the last event of the possession |
| possession.types | Goes through all events in this possession and adds the tags from ‘standart\_name’, ‘attack\_type\_name’. Returns a list. |
| possession.eventsNumber | Is set to unknown due to incompatibility between Instat and Wyscout Id’s |
| possession.eventIndex | Is set to 0 due to incompatibility between Instat and Wyscout Id’s |
| possession.startLocation.x | Is set to ‘pos\_x005F\_x' of the first event in the possession |
| possession.startLocation.y | Is set to ‘pos\_y' of the first event in the possession |
| possession.endLocation.x | Is set to ‘pos\_x005F\_x' of the last event in the possession |
| possession.endLocation.y | Is set to ‘pos\_x005F\_x' of the last event in the possession |
| possession.team.id | Is set to 0 due to incompatibility between Instat and Wyscout Id’s |
| possession.team.name | Is set to ‘possession\_team\_name' |
| possession.team.formation | Formations are extracted in preprocessing, more in the preprocessing section |
| groundDuel.opponent.id | Is set to 0 due to incompatibility between Instat and Wyscout Id’s |
| groundDuel.opponent.name | If ‘type.primary’ is duel, set to ‘opponent\_name’ else unkown |
| groundDuel.opponent.position | If ‘type.primary’ is duel, set to transformed ‘opponent\_position\_name’ else unknown |
| groundDuel.duelType | If ‘type.primary’ is duel, set to ‘dribble’ if dribble is in ‘type.secondary’, else either ‘defensive\_duel’ if it is in ‘type.secondary’ else ‘offensive\_duel’ |
| groundDuel.keptPossession | If ‘defensive\_duel’ is in ‘type.secondary’ set to unknown, else set to false if ‘possession\_team\_name’ is equal to Transition of possession else true |
| groundDuel.progressedWithBall | If ‘defensive\_duel’ is in ‘type.secondary’ set to unknown, else set to false if ‘possession\_team\_name’ is equal to Transition of possession else true |
| groundDuel.stoppedProgress | If ‘offensive\_duel’ is in ‘type.secondary’ set to unknown, else set to true if ‘possession\_team\_name’ is equal to Transition of possession else false |
| groundDuel.recoveredPossession | If ‘offensive\_duel’ is in ‘type.secondary’ set to unknown, else set to true if ‘possession\_team\_name’ is equal to Transition of possession else false |
| groundDuel.takeOn | If ‘type.primary’ is duel, set to true if ‘groundDuel.duelType’ is dribble else false |
| groundDuel.side | If ‘type.primary’ is duel, check for ‘pos\_y’ < 33.4 resp. >66.6 and set to left resp. right else set to None |
| groundDuel.relatedDuelId | Is set to 0 due to missing attribute |
| infraction.yellowCard | If yellow\_card is in ‘type.secondary’ set to true else false |
| infraction.redCard | If red\_card is in ‘type.secondary’ set to true else false |
| infraction.type | If ‘type.primary’ is infraction, if ‘action\_name’ is not deferred\_foul set to’regular\_foul’ else to ‘late\_card\_foul’ |
| infraction.opponent.id | Is set to 0 due to incompatibility between Instat and Wyscout Id’s |
| infraction.opponent.name | If ‘type.primary’ is infraction, set to ‘opponent\_name’ |
| infraction.opponent.position | If ‘type.primary’ is infraction, set to transformed ‘opponent\_position\_name’ |
| possession.attack.withShot | If at least one event from the same possession have a shot primary |
| possession.attack.withShotOnGoal | If ‘shot.onTarget’ of at least one event from the same possession is true |
| possession.attack.withGoal | If at least one event from the same possession include goal in secondary |
| possession.attack.flank | Is set to unknown |
| possession.attack.xg | Is set to unknown due to missing attribute |
| carry.progression | When carry is in ‘type.secondary’ calculate progress towards goal between the origin and destination coordinates |
| carry.endLocation.x | When carry is in ‘type.secondary’ set transformed ‘pos\_dest\_x005F’ |
| carry.endLocation.y | When carry is in ‘type.secondary’ set transformed ‘pos\_y’ |
| shot.bodypart | Transformed from ‘body\_name’ |
| shot.isgoal | If goal is in ‘type.secondary’ set to true else false |
| shot.onTarget | If ‘action\_name’ is either Shot on target or Goal |
| shot.goalZone | Is set to unknown due to missing attributes |
| shot.xg | Is set to unknown due to missing attributes |
| shot.postShotXg | Is set to unknown due to missing attributes |
| shot.goalkeeperActionId | Is set to unknown due to missing attributes |
| shot.goalkeeper.id | Is set to 0 due to incompatibility between Instat and Wyscout Id’s |
| shot.goalkeeper.name | Goalkeepers are extracted in preprocessing, more in the preprocessing section |
| aerialDuel.opponent.id | Is set to 0 due to incompatibility between Instat and Wyscout Id’s |
| aerialDuel.opponent.name | If ‘type.primary’ is duel, set to ‘opponent\_name’ else unkown |
| aerialDuel.opponent.position | If ‘type.primary’ is duel, set to transformed ‘opponent\_position\_name’ else unknown |
| aerialDuel.opponent.height | Is set to unknown due to missing attributes |
| aerialDuel.firstTouch | Is set to unknown due to missing attributes |
| aerialDuel.height | Is set to unknown due to missing attributes |
| aerialDuel.relatedDuelId | Is set to unknown due to missing attributes |

# Tag documentation

On the left hand side are the wyscout tags, on the right are the explanations how the tags are derived.

**Primary tags**

Every event has exactly one primary tag

clearance: if action\_name is “Clearance”

corner: if standart\_name is “Corner”

duel: if action\_name is includes “Tackle”, “Challenge”, “Air challenge”, “Successful dribbling”, “Unsuccessful dribbling”

free\_kick: if standard\_name is “Indirect free kick” or “Direct free kick”

game\_interruption: if action\_name is “Ball out of the field”

goal\_kick: if standard\_name is “Goal kick”

infraction: if action\_name is one of the words “Deferred foul”, “Foul”, “Yellow card”, “Red card”

interception: if action\_name is one of the words “Pass interceptions”, “Shots blocked”, “Cross interception”

offside:if action\_name is “Offside”

own\_goal: if action\_name is “Own goal”

pass: if action\_name is includes “Attacking pass accurate”, “Attacking pass inaccurate”, “Accurate key pass”, “Inaccurate key pass”, “Non attacking pass accurate”, “Non attacking pass inaccurate” and standard\_name is not a set piece

penalty: if standard\_name is “Penalty”

shot: if action\_name is includes “Blocked shot”, “Shot on target”, “Wide shot”, “Shot blocked by a field player”, “Shot into bar/post”, “Goal” and standard\_name is not a set piece

shot\_against: following event of a shot tag with a shot on target (or goal)

throw\_in: if standard\_name is “Throw in”

touch: if action\_name is “Dribbling” or Player has moved at least 5 Units (scaled by the Wyscout pitch coordinates) between receiving the ball and the next action

postmatch\_penalty: if the period is not regular playing time or extra time

postmatch\_penalty\_faces: always occurs after a postmatch\_penalty

not implemented: acceleration, fairplay, goalkeeper\_exit.

**Secondary tags**

secondary tags are always paired with one primary tag, a primary can have multiple secondaries

aerial\_duel: if action\_name is “Air challenge”

assist: if action\_name is “Assist”

back\_pass: if the primary is “pass” and the passing angle is >135 or <-135

ball\_out: if action\_name is “Ball out of the field”

carry: if action\_name is “Dribbling” or Player has moved at least 5 Units (scaled by the Wyscout pitch coordinates) between receiving the ball and the next action

conceded\_goal: if primary is “shot\_against” and the secondary in the event before includes the “goal”

counterpressing\_recovery: if requirements for “recovery” are fulfilled + the difference of the “matchTimestamp” of the current and previous event is <5 seconds

cross: if primary is “pass” and action\_name includes “Crosses accurate”, “Crosses inaccurate”, “Inaccurate blocked cross”

cross\_blocked: if primary is “pass” and action\_name is “Inaccurate blocked cross”

deep\_completed\_cross: if primary is “pass” and action\_name is “Crosses accurate” and the pass destination is within 20 meters of the goal.

deep\_completion: if primary is “pass” and action\_name includes “Accurate key pass” or “Attacking pass accurate” and the pass destination is within 20 meters of the goal.

defensive\_duel: if primary is “duel” and the possession\_team\_name is different from the team\_name of the current event

dribble: if action\_name is “Dribbling”

forward\_pass: if primary is “pass” and passing angle is <45 and >-45

foul: if action\_name is “Foul”

foul\_suffered: if secondary of the next event without the primary “duel” includes “foul” and team\_name from the future event and the current one is different

free\_kick\_cross: if primary is “free\_kick” and action\_name includes “Misplaced crossing from set piece with a shot”, “Accurate crossing from set piece with a shot”, “Misplaced crossing from set piece with a goal”, “Accurate crossing from set piece with a goal”, “Accurate crossing from set piece”, “Inaccurate set-piece cross”

free\_kick\_shot: if requirements for “shot” is fulfilled and primary is “free\_kick”

goal: if action\_name is “Goal”

ground\_duel: if primary is “duel” and action\_name is not “Air challenge”

hand\_pass: if primary is "pass” and body\_name is “Hand”

head\_shot: if primary is "shot” and body\_name is “Header”

key\_pass: if action\_name includes “Accurate key pass”, “Inaccurate key pass”

lateral\_pass: if primary is “pass” and secondary neither includes “back\_pass” or “forward\_pass” and has a pass length >12 meters

long\_pass: if primary is “pass” and pass length > 45 meters

loss: if primary does not include “infraction”, “game\_interruption”, “penalty” and either the possession\_status is “End” or action includes “Attacking pass inaccurate”, “Inaccurate key pass”, “Non attacking pass inaccurate”, “Lost balls”.

offensive\_duel: if primary is “duel” and secondary does not include “defensive\_duel”

opportunity: if action\_name includes “Chance was converted by”, “ Chance was not converted by”

pass\_into\_penalty\_area: if primary is “pass” and pass destination is in the penalty box

pass\_to\_final\_third: if primary is “pass” and pass destination is in the final thrid

penalty\_conceded\_goal: if secondary of the previous event includes “penalty\_goal”

penalty\_foul: if primary is infraction and primary of next event is “penalty”

penalty\_goal: if primary is “penalty” and “goal” is in the secondary

penalty\_save: if primary is “shot\_against” and previous event with primary “penalty” without “goal” in the secondary

progressive\_pass: if primary is “pass” and the pass distance is at least 30 meters closer to the opponent’s endline if the starting and finishing points are within a team’s own half or at least 15 meters closer to the opponent’s endline if the starting and finishing points are in different halves or at least 10 meters closer to the opponent’s endline if the starting and finishing points are in the opponent’s half

progressive\_run: if requirements of “carry” are fulfilled and carry end location is at least 30 meters closer to opponent endline if starting and finishing points are in own half or at least 15 meters closer to opponent endline if starting and finishing points are in different field halves or at least 10 meters closer to opponent goal if starting and finishing points are in opponent half

recovery: if the possession.team.name from the current event is different than the previous event and the current primary is not “game\_interruption”, “shot”, “shot\_against”

red\_card: if action\_name is “Red card”

save: if primary is “shot\_against” and previous event with primary “shot” and entry shot.isGoal is False

save\_with\_reflex: if action\_name is “Supersaves” and primary is “shot\_against”

short\_or\_medium\_pass: if primary is “pass” and “long\_pass” is not in the secondary

shot\_after\_corner: if primary is “shot” and matchTimestamp has a difference <14 seconds with a previous event with primary “corner”

shot\_after\_free\_kick: if primary is “shot” and matchTimestamp has a difference <14 seconds with a previous event with primary “free\_kick”

shot\_after\_throw\_in: if primary is “shot” and matchTimestamp has a difference <14 seconds with a previous event with primary “throw\_in”

shot\_assist: if action\_name includes “Misplaced crossing from set piece with a shot”, “Accurate crossing from set piece with a shot”, “Misplaced crossing from set piece with a goal”, “Accurate crossing from set piece with a goal”

shot\_block: if primary is “interception” and action\_name includes “Shot blocked by field player”, “Shots blocked”

sliding\_tackle: if action\_name is “Tackle”

touch\_in\_box: if action location is in the penalty box and primary is either “shot”, “pass” or “touch

yellow\_card: if action\_name is “Yellow card”

not implemented: dribbled\_past\_attempt, head\_pass, linkup\_play, loose\_ball\_duel, pressing\_duel, second\_assist, smart\_pass, third\_assist, through\_pass, under\_pressure, whistle.