Penalty Parsing documentation

1. Rules and templates
2. General Penalty Identifiers:

Keywords such as “penalized”, “foul”, and “illegal”, don’t indicate what type the penalty was.

1. Type-specific Penalty Rules

{penalty type: {

“inplay”: presnap / inplay,

“team”: offense / defense / unsure,

“line of measurement”: LOS / SOF, (line of scrimmage/spot of foul)

“yards”: 5/10/15,

“down”: 0 /1,

“identifiers”: [list of regex],

“offsettable”: boolean,

“declinable”: boolean}

}

1. Penalty result identifiers:

A penalty can be valid, offset, or declined. Each one is followed by a list of identifier regex.

1. Input

A TwoLine object defined in loop 2.

1. Algorithm
2. Check penalty exisitence
   1. If any genal penalty identifiers found, then return True
   2. If no genal penalty identifiers found, but found type-specific penalty identifiers, then also return True
3. Separate penalty part from the whole sentence
   1. Find the frontmost penalty identifier.
   2. Take everything after this penalty identifier as penalty part.
   3. Penalty part will be parsed separately from the rest of the sentence.
4. Extract penalty info
   1. Penalty type:
      1. Penalty type is matched by regex identifier.
      2. It is allowed to have multiple types of penalties.
      3. All other stats about a penalty (yards, line of measurement, down change) are looked up from type-specific penalty rules dictionary using the matched penalty type.
   2. Penalty result:
      1. Types are matched using regex
      2. If no obvious identifier for a penalty being offset and declined, then default valid.
      3. If multiple types of penalty was matched in the previous step, the result of each penalty is matched separately. Under 3 assumptions:
         * There can only be 1 valid penalty in each sentence.
         * If “offset” is matched at any time, both penalties in the same sentences can’t be valid.
         * One type of penalty can only happen once in a single play.
   3. Penalized team:

Usually the team penalized is pointed out in the penalty part of the play sentence. If the penalized team can be found, then use it. If the penalized team is not stated, use the offense/defense team from the type-specific penalty rules.

1. Output of Penalty Parser:

The output is a dictionary with such structure:

{

penalty\_type: {

“result”: valid/offset/declinded,

“team”: the abbreviation for offense/defense team of the play,

“yards”: the number of yards penalized,

“lom”: line of measurement. SOF/LOS,

“down change”: 0 (no change of down)/ 1(down plus 1)

},

…

(other penalty found in the same sentence, if exists)

}

1. How Penalty Parsing participates in the whole stat extraction process
2. After assigning player roles, execute penalty parsing module
3. Split penalty part of the play, if exists, and the rest of the play
4. Extract stats for non-penalty part of the play and calculate the predicted ending context
5. Parse the penalty part of the play
6. Update the predicted ending context using the output of penalty parsing
7. Use this version as the final predicted ending context and continue context validation.