

LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

**Hokeja pirmsspēles apskata automātiska veidošana**

**BAKALAURA DARBS**

Autors: **Kārlis Freimanis**

Stud. apl. kf11005

Darba vadītājs: Dr.sc.comp. profesors Guntis Arnicāns

RĪGA 2015

## ANOTĀCIJA

Hokeja pirmsspēles apskats, parasti ir ekspertu veidots rakstisks komandu salīdzinājums, kas balstās uz iepriekšējo spēļu statistiku un aktuālāko informāciju. Darbā aprakstīti būtiskākie aspekti automatiska pirmsspēles apraksta ģenerēšanā. Mērķis izveidot sistēmu, kas spēj izveidot saturīgus NHL pirmsspēles aprakstus. Darba rezultātā tika izveidota vienkārša sistēma, kas automātiski ievāc statistiku no NHL mājaslapas un spēj izveidot vienkāršus pirmsspēles apskatus.

Atslēgvārdi:

Dabiskās valodas ģenerēšana, hokejs, pirmsspēles apskats, NHL

## ABSTRACT

“Hockey game preview automatic generation”

Hockey game preview usually is expert written on statistics based two team comparison. In this work are reviewed most significant parts of hockey game preview generating process. Research target is to create automatic system that can generate meaningful NHL game previews. During the research was created system that automatically collects statistics from NHL home page and generate automated game previews.

Keywords:

Natural language generation, hockey, game preview, NHL

# SATURS

APZĪMĒJUMU SARAKSTS.....	6
IEVADS.....	7
1. DABISKĀS VALODAS ĢENERĒŠANAS SISTĒMA.....	9
1.1 Pielietojums.....	9
1.2. Vēsturiskas sistēmas.....	9
1.2.1 Kriminālstāstus ģenerējoša sistēma.....	9
1.2.2. Laika prognozi ģenerējoša sistēma.....	10
1.2.3. Zemestrīču uzraudzības sistēma.....	11
1.3. Teksta ģenerēšanas veidi.....	11
1.3.1. Statiska teksta.....	11
1.3.2. Sagataves aizpildīšana.....	11
1.4. Problēmas dabiskās valodas ģenerēšanā.....	12
1.6. Stāstu automātiskas ģenerēšanas sistēma izveide.....	12
2. UZDEVUMA ANALĪZE.....	14
2.1. Nozīmīgo faktu identificēšana.....	14
2.1. Statistikas ieguves metodes.....	14
2.1.1. Izvilkt datus no HTML.....	14
2.1.2. Atvērta koda datubāze.....	15
2.1.3. Pieeja līgas datubāzei.....	15
2.2. Datu kopas.....	15
2.2.1. Spēles protokols.....	15
2.2.2. Spēles kopsavilkums.....	15
2.2.3. Sezonas kopsavilkums.....	16
2.2.4. Sezonas kopsavilkums līdz konkrētam datumam.....	16
2.2.5. Izveidoto datu kopu salīdzinājums.....	16
2.3. Statistikas glabāšana.....	17
2.3.1. SQL datubāze.....	17
2.3.2. NOSQL datubāze.....	17
2.3.3. Teksta failos.....	17

2.3.4. Neglabāt datus.....	17
2.4. Tehnoloģiju izvēle.....	18
3. APSKATA ĢENERĒŠANA.....	19
3.1. Teksta ģenerēšana.....	19
3.2. Galvenie likumi datu pārbaudei.....	20
3.3. Likumu pārbaudes mehānisms.....	21
4. REZULTĀTU ATTĒLOŠANA.....	22
4.1. Epasts.....	22
4.2. Darbā izvēlētās attēlošanas metodes.....	22
4.2.1. Tīmeklī attēloti rezultāti.....	22
4.2.2. Audio formātā.....	22
5. IESPĒJAMIE SISTĒMAS UZLABOJUMI.....	24
5.1. Video apskats.....	24
5.2. Spēlei pieteikto spēlētāju saraksts.....	24
5.3. Spēlētāju un treneru pirmsspēles izteikumi.....	24
5.4. Vairāku valodu atbalsts.....	25
5.5. Teksta pārveidošana uz MP3 formātu uzlabošana.....	25
5.6. Sezonas un izslēgšanas spēļu atšķiršana.....	25
5.7. Attēlot tikai nozīmīgāko informāciju.....	25
5.8. Izveidot iespēju definēt likumus no pārlūka.....	26
5.9. NHL vēsturē zīmīgu statistikas radītāju sasniegšana.....	26
REZULTĀTI.....	27
SECINĀJUMI.....	29
LITERATŪRAS SARAKSTS.....	30
1. pielikums Pirmsspēles apskata paraugs (kā vienkāršs teksts) attēlotā spēle: <a href="http://www.nhl.com/gamecenter/en/preview?id=2014021151">http://www.nhl.com/gamecenter/en/preview?id=2014021151</a> .....	32
2. pielikums Sistēmas datubāzes shēma.....	34
3. pielikums Sistēmas uzstādīšanas pamācība.....	35

## APZĪMĒJUMU SARAKSTS

API – programmatūras bibliotēka, kurai ir definēts funkciju, datu struktūru klāsts, kas ļauj izmantot tās funkcionalitāti.

CSV - Teksta failu formāts kurā, katrs no ierakstiem ir atdalīts ar komatu vai semikolu.

Dabiskās valodas ģenerēšana – dabiskās valodas apstrādes uzdevums, kura procesā no datiem tiek iegūta dabiska valoda.

HTML - Vienkārša iezīmēšanas valoda, ko izmanto, lai izveidotu hiperteksta dokumentus, kas ir pārvietojami no vienas platformas uz citu platformu.

MySQL - Relāciju datubāžu vadības sistēma

NHL – Nacionālā hokeja līga.

PHP - atklātā pirmkoda skriptu valoda, kura sākotnēji bija paredzēta servera puses lietojumos dinamisku tīmekļa lapu ģenerēšanai.

SQL datubāze - Programmēšanas (strukturēta vaicājumu valoda) valoda, kas domāta, lai varētu apstrādāt datus relāciju datu bāzēs.

## IEVADS

Vairums cilvēku nemaz neapzinās, ka lielu daļu mūsdienās pieejamo rakstu ģenerē dators. 2011. gada 7. decembrī pētnieks Kris Hammonds izteica prognozi, ka aptuveni 2025. gadā 90% no visiem jaunajiem rakstiem ģenerēs dators[1]. Šāda prognoze varētu radīt iespaidu, ka pēc 10 gadiem 90% cilvēka rakstīto rakstu rakstīs mākslīgais intelekts, bet šāds pieņēmums būtu aplams. Kris Hammonds prognozē, ka attīstoties automātisku rakstu ģenerēšanas tehnoloģijām būtiski palielināsies mākslīga intelekta veidoto rakstu skaits. Par ko tad tiks ģenerēti raksti? Ir ļoti daudz mazo sporta līgu par kurām mūsdienās neraksta, bet nākotnē balstoties uz statistiku varētu tik veidoti plaši raksti par spēlētāju un komandu sniegumu. Sports nav vienīgā nozare kurā būtu iespējams izmantot automātiski ģenerētos rakstus. Dators ir spējīgs veidot rakstus jebkam, kam pie iepriekš definētiem ievaddatiem ir iespējams prognozēt kādam būtu jābūt rakstam.

Hokeja pirmsspēles apskats ir neatņemama spēles sastāvdaļa jau no pašiem sporta veida pirmsākumiem. Sākotnēji to veidoja paši spēlētāji balstoties uz redzēto un apskata mērķis bija tikai un vienīgi, lai izzinātu pretinieku un labāk sagatavotos spēlei.

Darba mērķis ir izpētīt, kā tiek veidotas automātisku rakstu ģenerēšanas sistēmas un izveidot sistēmu, kas spēj izveidot NHL hokeja spēlēm vienkāršu pirmsspēles apskatus latviešu valodā, kurā būtu aprakstītas komandu stiprās un vājās vietas. Izvēlējos NHL, kā eksperimenta objektu, jo NHL mājaslapā ir bagātīga un brīvi pieejama statistika[2].

Darba ietvaros veidotā sistēma varētu nodrošināt līgu ar kvalitatīviem pirmsspēles apskatiem, kas daļēji vai pilnībā varētu aizvietot cilvēku veikto darbu. Tomēr objektīvi skatoties, diez vai izdotos izveidot tik komplicētu un tik kvalitatīvu sistēmu, kas pārspētu profesionālus žurnālistus, bet tas nenozīmē, ka līdzīga tipa sistēmai nebūtu pielietojuma hokejā. Pasaulē ir tūkstošiem pus profesionālu un amatieru līgu par kurām neviens neraksta. Tieši šo līgu dalībnieki, manuprāt, būtu potenciālie šādu tipa sistēmu lietotāji. Automātiski ģenerēti raksti padarītu līgu bagātāku un radītu spēlētājos sajūtu, ka viņa līga nemaz tik ļoti neatpaliek no NHL.

Bakalaura darbs sastāv no 5 nodaļām. Pirmajā nodaļā vispārīgi tiek aprakstīts, kas ir dabiskās valodas ģenerēšanas un kāds ir pielietojums automātiski ģenerējamiem tekstiem. Otrajā, trešajā un ceturtajā nodaļā tiek aprakstīti automātisku rakstu ģenerēšanas sistēmas izveides posmi un izvēlētie risinājumi veidojot hokeja pirmsspēles apskatu ģenerēšanas sistēmu. Otrajām nodaļām tiek aprakstīts kā analizēt doto uzdevumu, identificēt būtiskākos datus un izvēlēties piemērotākās

tehnoloģijas. Trešajā nodaļā aprakstīts kā tiek ģenerēts raksts. Ceturtajā nodaļā tiek aprakstīts kā tiek noformētie sistēmas uzģenerētie ieraksti. Piektajā nodaļā tiek aprakstīti un analizēti dažādi iespējami sistēmas uzlabojumi.

# 1. DABISKĀS VALODAS ĢENERĒŠANAS SISTĒMA

Dabiskā valoda ir valoda kuru aktīvi lieto vai lietojusi kāda sabiedrība un kuras likumi izriet galvenokārt no tās lietošanas. Dabiskas valodas ģenerēšana ir process kura rezultātā datoram saprotama digitāla informācija tiek pārvērsta dabiskajā valodā, kas ir cilvēkam saprotams formāts.

## 1.1 Pielietojums

Automātiski ģenerētu rakstu stiprākā puse ir, iespēja, rakstu uzģenerēt acumirkļi un jebkurā diennakts laikā. Mūsdienu tehnoloģijas nenodrošina iespēju izveidot garus, komplicētus, saturīgus rakstus. Īsiem ziņojumiem automātiski ģenerētie raksti ir pietiekami efektīvi. Mūsdienās automātiski ģenerēti raksti dabiskajās valodās visplašāk tiek izmantoti sporta, ekonomikas un zinātnes nozarēs, jo šīm nozarēm ir nepieciešams tūlītējs rezultāts un ir plašas digitalizētas datubāzes kuras nepieciešams pārtulkot cilvēkam saprotamā formātā.

## 1.2. Vēsturiskas sistēmas

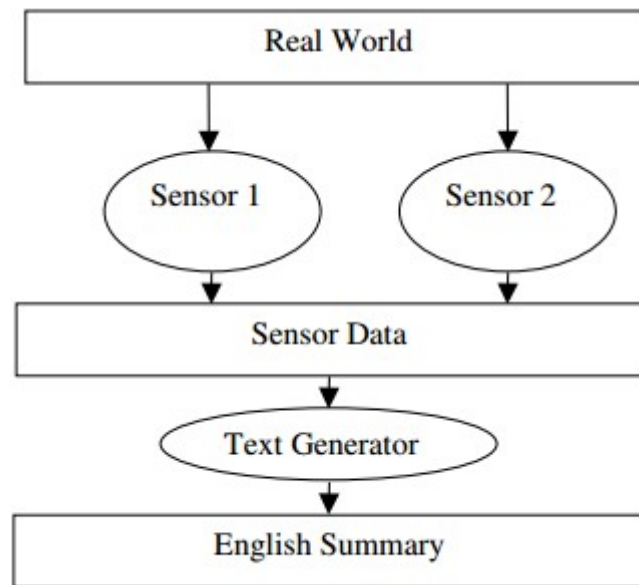
Pēdējās desmitgadēs ir izveidotas daudz dažādas sistēmas, kas ģenerē rakstu dabīgajā valodā. No tā var secināt, ka vismaz primitīvā līmenī dators ir spējīgs veidot cilvēkam noderīgus rakstus.

### 1.2.1 Kriminālstāstus ģenerējoša sistēma

Šeldons Kleins 1973. gadā izveidoja "Novel Writer system", kas ir viena no pirmajām ievērojamajām rakstus ģenerējošām sistēmām. Sistēma spēj izveidot aptuveni 2100 vārdu garus kriminālstāstus. Ievadā tiek padots apkārtējās vides apraksts, galveno lomu tēlu raksturojums. Stāsts tiek ģenerēts balstoties uz diviem algoritmiem 1. Likumu kopas kura bija iekodētas iespējamās izmaiņas no esošā stāvokļa uz nākamo. 2. Virkne iespējamo scēnu, kas bija atbilstošas esošajam tematam. Sistēmas stāsta veidošanā tika iekļauts arī nejaušības princips, līdz ar to pie vieniem ievaddatiem varēja tikt izveidoti dažādi stāsti. [3].

### 1.2.2. Laika prognozi ģenerējoša sistēma.

Pētnieki no Aberdīnas Universitāte 2005. gada 2. jūnijā publicēja pētījumu ar šokējošiem rezultātiem[4]. Pētījumā tiek apgalvots, ka ir izdevies izveidot sistēmu “SumTime-Mousam”, kas ģenerē kvalitatīvākas laika prognozes kā cilvēks spēj izveidot.[5]



1.1. att. “SumTime-Mousam” sistēmas darbības modelis.

Sistēma sastāv no vairākām daļām. Pirmkārt, tiek ievākti dati par esošajiem laika apstākļiem izmantojot sensoru iegūtos mērījumus, pēc tam sensoru iegūtā informācija tiek apstrādāta un izveidota laika prognoze. Pēdējais etaps ir digitalizēto laika prognozi pārvērsts dabīgās valodas tekstā.

1.1. tabula “SumTime-Mousam” ģenerētā teksta salīdzinājums ar cilvēka rakstīto[4]

Raksta autors	Raksts
cilvēks	S'LY 15-20 BECOMING 22-28 BY THIS EVENING. LATER VEERING S-SW 18-22
dators	S 16-21 BACKING SSE 21-26 BY MID AFTERNOON, THEN VEERING S BY EARLY EVENING AND SSW 18-23 BY MIDNIGHT

Rezultātā ģenerē īsu vienkāršu tekstu zinātniskā formātā, tāpēc varētu arī nepārsteigt fakts ka dators ir spējīgs uzģenerēt kvalitatīvāku rakstu par cilvēku. Tomēr, šādi pētījuma rezultāti liek aizdomāties. Vai jau nav pienācis tas mirklis kad dators cilvēku valodu pārvalda labāk par cilvēkiem.

### **1.2.3. Zemestrīču uzraudzības sistēma**

ASV žurnāls “Los Angeles Times” 2014 gada 17. martā šokēja žurnālistikas pasauli publicējot rakstu par zemestrīci. Zemestrīce spēka ziņā nebija ievērtības cienīga un Kalifornijā neierasts notikums. Uzmanību rakstam pievērsta publicēšanas laiks, kas bija 3 min pēc zemestrīces un ārpus žurnālistu darba laika, un raksta autors bija dators. Raksts bija īss vienkāršs un pietiekami kvalitatīvs, lai nevarētu pateikt, ka rakstu nav rakstījis cilvēks. Raksts pievērsa ļoti plašu interesi un raisīja diskusijas par to vai drīz žurnālistu aizvietos dators[6]. Par šo rakstu vēlāk tika publicēta virkne rakstu, citos nopietnos žurnālos kā piemēram “The Guardian”[7] un “BBC”[8] ziņojot, ka mūsdienās nav jābūt cilvēkam, lai varētu būt žurnālists.

### **1.3. Teksta ģenerēšanas veidi.**

Dabisko valodu ģenerēšanas sistēmas var būt dažādas sākot ar sistēmām, kas vienmēr atgriež vienādu rezultātu beidzot ar komplicētām sistēmām, kas veido pārdomātus gramatikas likumiem balstītus dabiskās valodas tekstus[9].

#### **1.3.1. Statiska teksta**

Vienkāršākā tipa teksta ģenerēšanas sistēma vienmēr izvada vienādu iepriekš definētu tekstu. Šādas sistēmas paraugs varētu būt brīdinājumu sistēma, kas paziņo par kādu notikumu[9].

#### **1.3.2. Sagataves aizpildīšana**

Izvadāmā teksta struktūra un katra struktūras elementa iespējamo vērtību kopa ir iepriekš definēta. Tipisks paraugs ir personalizēts epasts ar akcijas piedāvājumu[9].

### **1.3.3. Augsta līmeņa dabiskās valodas ģenerēšanas sistēma.**

Sistēma veido kontekstam atbilstošu, gramatikas likumiem atbilstošu tekstu. Izvēlas piemērotus vārdus aprakstāmajiem objektiem. Spēj veidot saliktus teikumus un veidot vairāku loģiski sekojošu teikumu virknes[9].

### **1.4. Problēmas dabiskās valodas ģenerēšanā**

Veidojot tekstus balstoties uz specificētiem ievaddatiem ir nepieciešams rast risinājumu uz sešām pamatproblēmām dabiskās valodas ģenerēšanā.

- Satura noteikšana – Izvēlēties kādu informāciju ir jāiekļauj tekstā un ko nepieciešams izlaist.
- Dokumenta strukturēšana – Izlemt kā organizēt un kādu veidot teksta struktūru.
- Leksikas izvēle – Izvēlēties konkrētus vārdus un frāzes, kas ir nepieciešamas, lai aprakstītu doto informāciju.
- Teikuma veidošana – Izvēlēties, kā izlemt kādu informācijas apjomu iekļaut vienā teikumā.
- Atsaukties uz informācijas objektiem – Izlemt kādus objekta parametrus izmantot, lai atsauktos uz informācijas objektu.
- Teikumu apvienošana – Noteikt, kā izcelt un apvienot tekstā teikumus kuriem būtu jāsasaista un izmantojot gramatiski korektus teikumus, izteikt vēlamo domu.

Pieminētās sešas problēmas nav vienīgās ar kurām saskaras veidojot dabiskās valodas tekstus. Sarakstu varētu vēl turpināt ar vairākām problēmām kā, piemēram, ar paragrāfu strukturēšanu. Tomēr šīs sešas problēmas ir lielisks veids, kā apkopot galvenās problēmas, uzdevumus, ko nepieciešams atrisināt veidojot, ģenerējot tekstu dabīgajā valodā[10].

### **1.6. Stāstu automātiskas ģenerēšanas sistēma izveide.**

“Quill” ir viena no retajām platformām, kas veidota ar mērķi izveidot sistēmas, kas automātiski ģenerē dabīgās valodas tekstus. Uz šīs platformas bāzētu sistēmas izstrāde tiek iedalīta trīs daļās[11].

- 1) Uzdevuma analīzes daļa – Izvērtē pieejamo informāciju, identificē būtiskākos faktus, informācijas kopas. Ne visa digitālā informāciju ir nepieciešams pārtulkot.

- 2) Tekstu ģenerēšanas daļa – Definē likumus, kas nosaka kad nepieciešams ģenerēt tekstu. Izveido plānu kā strukturēt ģenerējamo tekstu.
- 3) Informēšanas daļa – Noskaidro kādā formātā jābūt sistēmas izvaddatiem, dara tos pieejamus sistēmas lietotājiem.

Pēc līdzīga principa veidoju arī hokeja pirmsspēles apskatu ģenerēšanas sistēmu. Vispirms, izvērtēju iespējamās datu kopas no kurām ir iespējams veikt loģisko spriešanu un veidot teikumus. Otrajā daļā tika definēti likumi, kas nosaka kad ir konstatēts ziņojuma vērts notikums un vajadzības gadījumā ģenerēts teksts. Trešajā daļā izvērtētas iespējas kā būtu iespējams noformēt uzģenerētos rezultātus, lai sasniegtu maksimālu efektu.

## 2. UZDEVUMA ANALĪZE

Pirmais solis automātiski ģenerēta teksta izveidei, ir ievaddatu un pieejamo datu kopu analīze, kurā jāatšķir būtisko informāciju no nebūtiskās un jāizlemj kā strukturēt datus efektīvākai tekstu ģenerēšanai.

### 2.1. Nozīmīgo faktu identificēšana

Labākais veids kā noteikt kāda informāciju, ko nepieciešams iekļaut automātiski ģenerējamā tekstā ir analizējot esošos cilvēku radītos tekstus.

Izvērtējot hokeja pirmsspēles apskatus var secināt, ka klasisks pirmsspēles apskats ietver sevī atbildi uz četriem jautājumiem.

- Kāds ir komandu sniegums pēdējās spēlēs?
- Kura komanda bija labāka iepriekšējās savstarpējās spēlēs?
- Cik veiksmīga sezona ir komandām?
- Kuri spēlētāji ir ievērtības cienīgi?

Balstoties uz šiem jautājumiem tika izlemts, ka nepieciešams iegūt statistiku, par hokeja spēlēm vismaz sezonas apmērā un jāapkopo komandu un spēlētāju statistika.

### 2.1. Statistikas ieguves metodes

Bez ievaddatiem nav iespējama dabīgas valodas tekstu ģenerēšana, tāpēc pirmais uzdevums ir apzināt kā iespējams iegūt uzdevuma mērķim atbilstošākos ievaddatus.

#### 2.1.1. Izvilkt datus no HTML

Projekta ietvaros izvēlējos datus iegūt no NHL oficiālās mājaslapas, jo mājaslapā ir pieejam visaktuālāk un visprecīzākā statistika par NHL. HTML lapas apstrādes lielākais mīnus, ir risks ka laika gaita varētu mainīties HTML struktūra līdz ar to datu ieguve mehānisms varētu kļūt pilnīgi nederīgs.

### **2.1.2. Atvērtā koda datubāze**

Atvērtā koda datubāzes ir lielisks veids, kā varētu iegūt jau strukturētus datus. Tomēr atvērtajiem projektiem bieži trūkums ir ka dati netiek atjaunoti, vai ir nepilnīgi. Manām vajadzībām piemērotākā datubāze ko izdevās atrast bija "Open Source Sports"[12], no kuras bija iespējams lejuplādēt samērā plašu statistiku par NHL CSV formātā. Tomēr statistika bija pieejama tikai līdz 2011. gadam, kas neatbilda prasībām, jo nenodrošināja iespēju iegūt aktuālāko informāciju. Jāņem vērā arī to ka atvērtā koda

### **2.1.3. Pieeja līgas datubāzei**

Ideālā gadījumā būtu pieeja hokeja līgas datubāzei, šādā gadījumā nebūtu jābaidās, ka laika gaitā varētu, kas mainīties un izvēlētais mehānisms pārstatu darboties.

## **2.2. Datu kopas**

Teorētiski būtu iespējams veikt secinājumus no primitīvākās datu kopas, kas runājot par hokeja spēli ir spēles protokols. Tomēr šāda pieeja būtu gaužām lēna, prasītu daudz procesora jaudas, palielinātu iespēju kļūdīties rakstot secināšanas likumus. Tādēļ nepieciešams izveidot kopsavilkumus, kas dotu iespēju rakstīt vienkāršākus efektīvākus secināšanas likumus.

### **2.2.1. Spēles protokols**

Spēles protokols ir primitīvākā hokeja spēles statistikas datu kopa, kurā ir pieejama informācija par visiem notikumiem kas notikuši spēles laikā un dotie notikumi ir sakārtoti pēc laika, kad tas ir noticis. Spēles protokols ir noderīgs likumos kuros ir svarīga secība kāda notikuši notikumi spēlei gaitā, piemēram, komanda A guvusi pirmā vārtus n spēles pēc kārtas. Spēles protokoli ir neefektīvi, ja jāveic vispārīga spriešana par spēles un sezonas statistiku.

### **2.2.2. Spēles kopsavilkums**

Spēles kopsavilkumā tiek summēti spēles protokolā minētie notikumi, tādā veidā tiek iegūta datu kopa, kas raksturo notikuma biežumu spēles laikā. Šāda datu kopa ir noderīga, ja nepieciešams izveidot likumu, kas efektīgi izvērtē notikuma biežumu spēļu laikā, piemēram, komanda ir

uzvarējusi iemetienos  $n$  no  $m$  pēdējām spēlēm.

### 2.2.3. Sezonas kopsavilkums

Sezonas kopsavilkumā tiek summēti visu spēļu kopsavilkumos minētie notikumi un iegūts kopējais notikuma biežums sezonā. Šāda datu kopa ir noderīga, lai varētu efektīvi noteikt komandas sniegumu sezonas mērogā. Piemēram, komanda ieņem  $n$ . vietu līgā.

### 2.2.4. Sezonas kopsavilkums līdz konkrētam datumam

Sezonas kopsavilkums satur summu par visiem notikumiem spēlēs līdz kādam noteiktam datumam. Šāda datu kopa nepieciešama, lai varētu ātri salīdzināt komandas šīs sezonas sniegumu ar citām sezonām. Piemēram, Komandas A uzvarējusi vairāk spēles kā pēdējās  $n$  sezonās.

### 2.2.5. Izveidoto datu kopu salīdzinājums

Dažādas datu kopas tiek veidotas, lai varētu efektīvāk atbildēt uz dažāda tipa jautājumiem. Zemāk redzamajā tabulā tiek veikts aptuvenš aprēķins, cik vienības datus būtu jāapskata izmantojot katru datu kopu lai iegūtu atbildi uz uzdoto jautājumu. Protams jāņem vērā, ka ne visas izveidotās datu kopas spēs sniegt atbildi uz uzdotajiem jautājumiem.

#### 1.2.1 . datu kopu salīdzinājums

Jautājums	Spēles protokols	Spēles kopsavilkums	Sezonas kopsavilkums	Sezonas kopsavilkums līdz konkrētam datumam
Vai komanda A guva 1. vārtus pēdējā spēlē?	1 līdz $p$			
Vai komanda A uzvarēja pēdējo spēli?	$p$	1		
Vai komanda A ir pēdējā vietā līgā.	$p*s*k$	$s*k$	1	
Vai komandai A šī sezona ir veiksmīgāka par iepriekšējo?	$p*s*k*2$	$s*k*2$	$s*k+1$	1

p – protokola ierakstu skaits

s – spēļu skaits sezonā

k – komandu skaits līgā

## **2.3. Statistikas glabāšana**

Tikpat svarīgi kā izvēlēties pareizu struktūru kā glabāt datus ir izvēlēties piemēru datu glabāšanas sistēmu, kas nepieciešama, lai varētu veikt efektīvu spriešanu ar datiem.

### **2.3.1. SQL datubāze**

Projekta ietvaros datu glabāšanai tika izvēlēta, MYSQL datubāze, jo SQL tipa datubāzes pasaulē ir populārākas informācijas glabāšanai[13]. Izmantotas tehnoloģijas popularitāte garantē, ka veiktais risinājums būs saprotams vairāk cilvēkiem kā arī garantē dotās tehnoloģijas kvalitāti un spēju pielāgoties universālam uzdevumam.

### **2.3.2. NOSQL datubāze**

NOSQL tipa datubāze arī varētu būtu gana laba izvēle datu glabāšanai, jo šī uzdevuma ietvaros netiek glabāti netipiski dati.

### **2.3.3. Teksta failos**

Teorētiski informāciju varētu glabāt arī teksta failos, tomēr šāds variants nenodrošinātu efektīvu spriešanu, jo teksta failu apstrāde ir salīdzinoši lēna. Teksta formāts varētu būt apsverams kā rezerves kopiju formāts.

### **2.3.4. Neglabāt datus**

Šāds variants ir pilnīgi nepieņemams, ja tiek izmantots variants, ka statistika tiek iegūta apstrādājot HTML lapas. Šāda pieeja kritiski samazinātu spriešanas ātrumu, jo būtu nepieciešams lejuplādēt HTML dokumentus no ārēja resursa, kā arī vajadzīgās informācijas izvilkšana no HTML

dokumenta, kas prasītu gan procesora jaudu, gan operatīvo atmiņu, gan palielinātu tīkla noslodzi. Sliktākajā gadījumā daudzie pieprasījumi no sistēmas puses radītu iespaidu, ka sistēma ir uzbrucējs, kas cenšas pārslogot statistikas uzturēta mājaslapu un sistēmai tiktu liegta pieeja informācijas resursam.

## 2.4. Tehnoloģiju izvēle

Izvēlētajām tehnoloģijām ir nepieciešams būt brīvi pieejamām, labi dokumentētām un mūsdienīgām. Izvēlētajam risinājumam, jānodrošina:

- Datu ievākšanas moduļa izveidi
- Teksta ģenerēšanas moduļa izveidi
- Rezultātu attēlošanu

Programmēšanas valoda tika izvēlēts PHP, kā vienai no populārākajām tīmekļa lapu izrādei izmantotajam programmatūrām[14]. PHP ir iekļautas visas mūsdienīgās programmēšanas tehnikas. Kā PHP ietvars tika izvēlēts Laravel 5[15]. Projekta sākumā Laravel 5 vēl nebija stabilas versijas iepriekšējo Larvel ietvaru versiju kvalitāte iedrošināja riskēt uz izvēlēties viss jaunāko Larvel ietvaru kāds ir pieejams. Izmantojot Laravel ir izveidoti visi trīs augstāk minētie moduļi – datu ievākšana, teksta ģenerēšana un rezultātu attēlošana.

Kā jau 2.3. nodaļā minēts datu glabāšanai tik izvēlēts MYSQL, jo MYSQL ir plašāk izmantotā bezmaksas SQL tipa datubāze.

### 3. APSKATA ĢENERĒŠANA

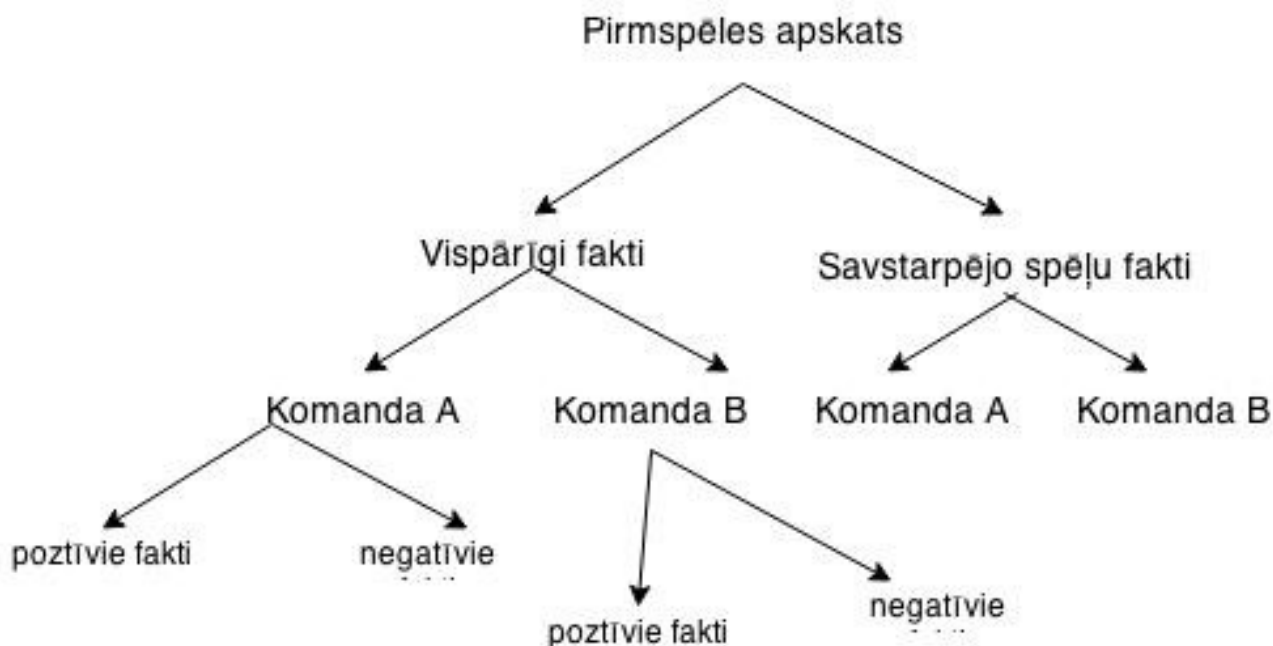
Apskata ģenerēšanas daļa tiek aplūkots kā tiek analizēti datoram pieejami dati un veidots dabīgās valodas teksts.

#### 3.1. Teksta ģenerēšana

Meklējot atbildes uz 1.4. nodaļā aprakstītajam dabīgās valodas ģenerēšanas pamatproblēma tiek veidots Teksta ģenerēšanas plānojums.

Satura noteikšana – Apskatā tiks iekļauta informācija par sekojošiem statistikas rādītājiem komandām: uzvaras, zaudējumi, spēlētiem vairākumiem, vairākumos gūtajiem vārtiem, izdarītajiem spēka paņēmieniem, uzvarētiem iemetieniem, piespiestajām un nepiespiestajām kļūdām, bloķētiem metieniem, soda minūtēm.

Dokumenta strukturēšana – Katrs teikums, jeb būtiskais fakts ietilps vienā no sešām grupām. Vizuālajā attēlojumā vienas grupas elementi tiks attēloti vienkopus.



3.1. att. Apskata struktūra

Kā attēla 3.1. redzams teksts tiek dalīts divās grupās vispārīgie fakti un savstarpējo spēļu

fakti. Katrai no šīm grupām ir divas apakšgrupas jeb fakti par konkrēto komandu. Zem vispārīgajiem faktiem, katras komandas fakti dalās vēl divās apakšgrupās, pozitīvie fakti un negatīvie, bet savstarpējo spēļu fakti netiek dalīti pēc līdzīga principa, jo vienas komandas pozitīvie fakti ir otras komandas negatīvie fakti, kas radītu lieku teksta dublēšanos.

Leksikas izvēle – Vārdi un frāzes tiek izvēlēti subjektīvi un atbilstoši darba autora zināšanām par hokeju, jo hokeja terminoloģijas izpēte netiek uzskatīta par būtisku.

Teikuma veidošana – Teikumā tiek iekļauta informācija par vienu hokeja statistikas faktu.

Atsaukties uz informācijas objektiem – Sistēmā vienīgais komplicētais objekts kura raksturošanu būtu vērts raksturot šajā sadaļā ir hokeja spēle. Hokeja spēli galvenie raksturojošie parametri ir spēles laiks un spēlējošās komandas.

Teikumu apvienošana – Teikumi sistēmas prototipa ietvaros netiks apvienoti.

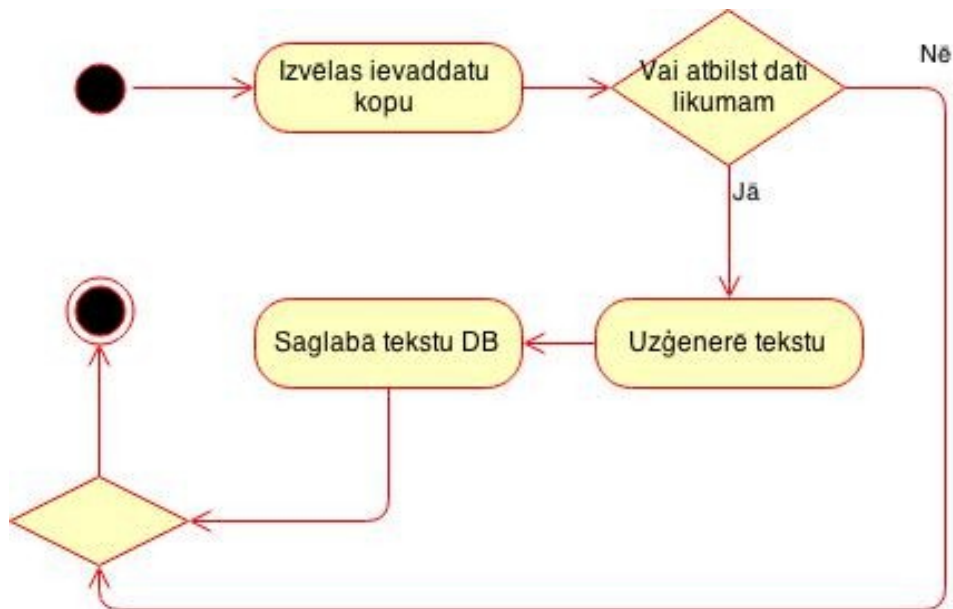
### **3.2. Galvenie likumi datu pārbaudei**

Apskata ģenerēšanas ietvaros likums ir datu pārbaude, kas var iniciēt dabīgā valodas teksta ģenerēšanu. Likumi tiek veidoti ar domu pārbaudīt vai pieejamos datos nav kāds nozīmīgs fakts, ja fakts tiek konstatēts, nepieciešams iniciēt atbilstoša teksta izveidošanu.

- Sērija – pārbauda vai kāds notikums ir atkārtojies vairākas reizes pēc kārtas, piemēram komanda izcīnījusi  $n$  uzvaras pēc kārtas. Par sēriju tiek uzskatīts notikums kur  $n \geq 3$ .
- Lielisks sniegums – Pārbauda vai komanda ir bijusi īpaši veiksmīga kādā rādītājā pēdējās spēlēs, piemēram komanda ir bijusi labāka iemetienos  $n$  no  $m$  pēdējām spēlēm.  $m \geq 5$  un  $n/m \geq 0.8$ .
- Rādītājs ir zīmīgs līgas ietvaros – Komanda ir starp 5 labākajam vai 5 sliktākajam komandām līgā dotajā statistika rādītājā.
- Rezultatīvākais spēlētājs komandā – Nosaka kurš no spēlētājiem pēdējās spēlēs ir bijis visrezultatīvākais.

### 3.3. Likumu pārbaudes mehānisms

Likumu pārbauda funkcija kurai ievadā tiek padots: spēles identifikators, komandas identifikators, pārbaudāmais statistikas rādītājs, sagatavota datu kopa.



3.2. att. Likumu pārbaudes mehānisms

Visi likumi tiek veidoti pēc vienota principa. Likumam tiek padota sagatavoti ievaddati, ja ievaddati atbilst likumā definētajām prasībām tiek piemeklēts piemērota sagatave un izveidots teksts atbilstoši iegūtajai informācijai, rezultātā uzģenerētais teksts tiek saglabāts datubāzē.

## 4. REZULTĀTU ATTĒLOŠANA

Komerčiālām dabiskās valodas tekstu ģenerēšanas sistēmām, bieži vien formāts kādā tiek pasniegts uzģenerētais teksts ir svarīgāks, kā teksta kvalitāte.

Rezultātu attēlošanai daļa atbild uz šādiem pamatjautājumiem.

- Kas drīkst piekļūt uzģenerētajai informācijai?
- Kādā formātā datu attēlošana vislabāk būs saprotama informācijas lietotājam?

### 4.1. Epasts

Viens no populārākajiem veidiem kā izplatīt automatiskos tekstus ir epasts. Diemžēl neizdevās atrast pētījumu kurā būtu precīzi aprēķināts cik procentus no visiem epastiem ģenerē dabiskās valodas ģenerēšanas sistēmas, bet izvērtējot manu personīgo epastu 3 no aptuveni 150 saņemtajam ziņām ir cilvēka rakstītas, kas parāda to cik būtiska loma epastam ir automatiski ģenerētu tekstu izplatīšanai.

### 4.2. Darbā izvēlētās attēlošanas metodes

Darba laikā automatiski ģenerētos tekstu nav plānots no neviena slēpt tādēļ datu attēlošanai nepieciešams izvēlēties risinājumu, kas jebkuram varētu nodrošināt brīvu pieeju sistēmai, jebkurā laikā.

#### 4.2.1. Tīmeklī attēloti rezultāti.

Kā piemērotākais risinājums publiski brīvi pieejamai datu attēlošanai tika izvēlēts attēlot rezultātus tīmekļa lapā izmantojot HTML.

#### 4.2.2. Audio formātā

Apskata lasīšana prasa koncentrēšanos un nedod iespēju šai darbībai paralēli veikt citus uzdevumu. Hokeja pirmsspēles apskata apskatīšana noteikti nav svarīgākais ko cilvēks varetu gribēt darīt. Tāpēc, tika domāts, kā varētu padarīt to vieglāk pieejamu cilvēkam. Rezultāta tika secināts, ka

uzģenerēto tekstu varētu pārvērst audio formātā. Šāda pieeja dotu iespēju izveidoto apskatu klausīties fonā kā mūziku un veltīt tam tikai daļītu uzmanību, vai arī varētu apskatu klausīties kā radio izmantojot telefonu, pābraucīenu laikā.

Ideja viennozīmīgi ir ievēribas vērtā un varētu pielīdzināt izveidoto sistēmu hokeja ziņu radiostacijai. Tomēr meklējot risinājumus nācās secināt, ka ideju izmantojot bezmaksas rīkus šobrīd nav iespējams realizēt pieņemamā kvalitātē.

Labākais pieejamais rīks latviešu valodas pārveidošanai audio formātā, kas pētījumā tika atrasts ir Google tiešsaistes teksta pārveidotājs audio formātā[16]. Diemžēl Google ģenerētajiem audio failiem ir būtiskas nepilnības. Lielākā no tām ir vienmuļa teksta veidošana, bez jebkādas kontekstam pielāgotas intonācijas. Otrs būtiskākai trūkums, sintezētā balss izklausās nedabīga, mehāniska.

Tas, ka latviešu valodā nav izveidota sistēma, kas tekstu pārveido augstas kvalitātes audio formātā nav pārsteigums, jo latviešu valodu lieto salīdzinoši mazs cilvēku skaits. Tāpēc pārbaudīju kā šīs tehnoloģijas strādā angļu valodā. Google bezmaksas piedāvātais risinājums arī neradīja uzticību, jo līdzīgi latviešu valodas versijai arī angļu valodā balss mehāniska un nepielāgojas kontekstam. Tomēr, tik atrasts maksas risinājums "Ivona", kas tiesām spēja radīt augstas kvalitātes runu audio formātā, gan balss neizklausījās mehāniska, gan runa veidota kontekstam atbilstoša[17].

Tas, ka eksistē ļoti efektīvas cilvēka runas sintēzes sistēmas angļu valodā nozīme to, ka tuvāko gadu laikā varētu tikt izveidots līdzvērtīgs bezmaksas risinājums un agrāk vai vēlāk arī latviešu valodai izveidos augsta līmeņa audio sintēzes tehnoloģijas.

## 5. IESPĒJAMIE SISTĒMAS UZLABOJUMI

Nodaļā tiek uzskaitīt un izvērtēti dažādi iespējamie bakalaura darbā izstrādātās sistēmas uzlabojumi.

### 5.1. Video apskats

Efektīvs papildinājums spēles apskatam, varētu būtu video apskats kurā tiktu attēloti spilgtākie notikumi pēdējās spēlēs. Projekta sākumā iespēja veidot video apskatu likās utopiska un tika atmesta kā praktiski nerealizējama.

Pētot un apdomājot iespējamus risinājumus kā automātiski no video varētu iegūt spilgtākos momentus tika secināts, ka šāda ideja nemaz nav tik utopiska, un, ja būtu pietiekami daudz laika šāda ideja būtu pat īstenojama.

Pētnieki no Dublinas pilsētas Universitāte jau 2002. gadā publicējuši rakstu par to kā automātiski apstrādājot sporta video iespējams iegūt labākos momentus. Galvenā ideja ir ka spēles komentētāju runas stils būtiski izmainās mirklī kad laukumā notiek īpašs moments[18].

### 5.2. Spēlei pieteikto spēlētāju saraksts

Esošās sistēmas būtiskākais trūkums ir ka tā nezina kādi spēlētāji piedalīsies spēlē, tomēr klasiskā pirmsspēles apskatā vienmēr ir pieminēts, ka svarīgs komandas spēlētājs nepiedalīsies spēlē, vai, ja kādam spēlētājam šī būs pirmā spēle. Uzlabojums nebūtu sarežģīts. Viss, kas nepieciešams ir atrast informācijas resursu no kura iegūt šādu statistiku, izveidot jaunu likumus, kas pārbauda spēlētāju sarakstus un izveidot jaunas teksta veidnes atbilstoši likumu pārbaudēm.

### 5.3. Spēlētāju un treneru pirmsspēles izteikumi

Preses konferences un Twitter ierakstos rakstīti komentāri bieži vien ir cilvēku radītu hokeja pirmsspēļu apskata sastāvdaļa. Diemžēl šādu papildinājumu pievienot sistēmai nepavisam nebūtu vienkārši. Pirmkārt, internets ir pilns ziņu un sasaistīt spēlētāju ar konkrētu rakstu nav viegls uzdevums. Tomēr, ja būtu risinājums, kā sasaistīt spēlētājus ar piemēram Twitter kontiem, tad šādu jaunumu pievienošana, jau būtu pavisam reāla, jo tad atliktu izpētīt lietotāja ziņas ierobežotā laika

periodā un meklēt ziņas atslēgvārdus, kas varētu liecināt, ka ir pieminēta aprakstāmā spēle.

#### **5.4. Vairāku valodu atbalsts**

Pat ja izdotos sistēmu izveidot tik labu, ka teksti tiktu ģenerēti līdzvērtīgi vai labāki par cilvēku rakstījumiem, Latvijas tirgus ir pārāk mazs, lai šāda sistēma gūtu būtiskus panākumus. Tāpēc viens no būtiskākajiem sistēmas uzlabojumiem būtu vairāku valodu atbalsta ieviešana. Balstoties uz esošo veidņu aizpildīšanas funkcionalitāti ieviest citas valodas nebūtu nekādu problēmu. Tomēr ja kļūtu sarežģītāks dabiskās valodas ģenerēšanas process būtiski sarežģītāks kļūtu arī jaunu valodu pievienošanas process, jo katrai valodai ir savi likumi, un izveidot likumu kopu, kas korekti veidotu komplicētus strukturētus datus jau vienai valodai ir ļoti darbietilpīgs darbs.

#### **5.5. Teksta pārveidošana uz MP3 formātu uzlabošana**

Pirmais variants izveidot konkurētspējīgu programmatūru teksta pārveidošanai uz audio formātu latviešu valodai. Tomēr šāda pieeja prasītu daudz resursu un piepūles.

Būtiski efektīvāks būtu variants, ja līdzīgi kā tekstu arī audio formātam tiktu izveidotas savas veidnes kurās tiktu korekti un kontekstam atbilstoši ierunāts teksts. Ar šādu metodi neiegūtu universālu risinājumu, bet iegūtu uzdevumam piemērotu ātro risinājumu, kas būtiski uzlabotu esošo funkcionalitāti.

#### **5.6. Sezonas un izslēgšanas spēļu atšķiršana**

Ievāktajā informācijā sezonas spēles tiek atšķirtas no izslēgšanas spēlēm tomēr neviens likums neņem vērā šo atšķirību. Normālā pirmsspēles apskatā noteikti tiktu īpaši atzīmēta sezonas pirmā spēle vai izslēgšanas spēļu izšķirošā spēle.

#### **5.7. Attēlot tikai nozīmīgāko informāciju**

Esošajā spēles apskatā tiks parādīts ikviens fakts kas tiks uzģenerēts vienalga cik liels arī kopējais apskats neveidotos. Vienkāršākais risinājums šai problēmai būtu ieviest maksimāli parādāmu faktu daudzumu un izveidot statistiku par to cik bieži katrs fakts ir unikāls.

## **5.8. Izveidot iespēju definēt likumus no pārlūka**

Esošajā risinājumā likumi un ģenerētais teksts tiek iekodēti programmatūras tekstā. Varētu izveidot risinājumu, kas dod iespēju analizēt pieejamo statistiku un dot lietotājam iespēju piešķirt atbilstošu tekstu pie likuma veiksmīgas izpildes. Šāda funkcionalitāte būtiski uzlabotu sistēmas pielāgojamību dažādu lietotāju prasībām, kā arī atvieglotu sistēmas pielāgošanu citām līgām un sporta veidiem.

## **5.9. NHL vēsturē zīmīgu statistikas rādītāju sasniegšana.**

Pārbauda vai gaidāmajā spēlē komandas un spēlētāji nav tuvu rekorda sasniegšanai vai zīmīgam skaitlim kāda no statistikas rādītājiem.

## REZULTĀTI

Bakalaura darbam izvirzītais mērķis ir daļēji sasniegts. Sistēma ir spējīga veikt tai sākumā uzstādītas prasības. Tiek automātiski iegūti dati un balstoties uz iegūtajiem datiem tiek automātiski veidots spēles apskats, tomēr izveidotais apskats ir pārāk primitīvs un ne tuvu nav spējīgs konkurēt ar cilvēku veidoto, tāpēc nevar uzskatīt, ka uzdevums ir pilnībā izpildīts.

Pirmais un, iespējams, komplicētākais uzdevums bija ievākt statistiku par NHL spēlēm. Projekta sākumā tika meklētas, jau gatavas datubāzes vai viegli izmantojams API NHL statistikas ieguvei. Izdevās atrast dažādas datu krātuves ar NHL statistiku. Ievērojamāka no tām "Open Source Sports"[12]. Tomēr datubāze bija novecojusi un netika atjaunināta kas neatbilda darba mērķim.. Tika izlemts, ka nepieciešams veidot datu ieguves mehānismu. Sistēma statistiku ievāc no NHL mājaslapas apstrādājot HTML. HTML ērtākai apstrādei tiek izmantots Laravel paplašinājums "HtmlDom"[19]. Sākotnējais plāns bija ievākt statistiku no 2005. gada sezonas, tomēr tika secināts, ka tik senām sezonām datus ievākt nav vērts, jo vairākām spēlēm statistika NHL mājaslapa nav pieejama. Uzdevuma vienkāršošanai tik izlemts iegūt statistiku sākot no 2013. gada rudens. Izveidotais risinājums, dod iespēju neveidot statistikai rezerves kopijas, jo visu nepieciešamo statistiku ir iespējams lejuplādēt no NHL mājaslapas. Statistikas ieguves mehānisma ātrdarbība, nav lieliska, bet pilnīgi atbilstoša uzdevuma. Pilna 2 sezonu statistikas ievākšana var ilgt līdz 12 stundām, vienas dienas statistika no 1 sekundes līdz 2 minūtēm. 12 nav daudz, jo šāds imports ir jāveic tikai vienreiz sistēmas lietošanās vēsture un 2 minūtes nav daudz jo datu atjaunināšana notiek vienu reizi dienā. Laikietilpīgākais solis ir HTML ieguve no NHL mājaslapas, ko optimizēt varētu tikai uzlabojot interneta savienojuma ātrumu, bet pat ar labāko interneta savienojumu, datu ieguvi bremsētu NHL mājaslapas serveris. Kopumā varētu teikt, ka datu ieguves moduļa izveide ir izdevusies, jo ir izdevies iegūt vajadzīgo statistiku un sistēma ir spējīga automātiski ievākt aktuālāko informāciju.

Otrs lielākais projekta uzdevums ir uzģenerēt pirmsspēles apskatu. Tika izstrādāti četri pēc būtības atšķirīgi algoritmi.

- Algoritms, kas meklē aktīvās sērijas komandām, jeb vai komanda vairākas spēles pēc kārtas nav bijusi labāka par pretinieku. Algoritms pārbauda sekojošu komandu statistiku pēdējās spēlēs: gūtie vārti, vairākumu skaits, veiksmīgie vairākumi, mazākumu skaits, ielaisti vārti mazākumā, spēka paņēmieni, piespiestās kļūdas, nepiespiestās kļūdas, bloķētie metieni,

soda minūtes, izdarīto metienu skaits pa periodiem un spēlē kopsummā.

- Algoritms, kas meklē vai kādā no spēles rādītājiem komanda bieži(80% gadījumu) pēdējās spēlēs ir bijusi labāka par pretinieku. Algoritms pārbauda sekojošu komandu statistiku pēdējās spēlēs: gūtie vārti, vairākumu skaits, veiksmīgie vairākumi, mazākumu skaits, ielaisti vārti mazākumā, spēka paņēmieni, piespiestās kļūdas, nepiespiestās kļūdas, bloķētie metieni, soda minūtes, izdarīto metienu skaits pa periodiem un spēlē kopsummā.
- Algoritms, kas meklē komandā pēdējās spēlēs rezultatīvākos spēlētājus. Algoritms pārbauda sekojošu spēlētāju statistiku pēdējās spēlēs: vārti, piespēles.
- Algoritms, kas nosaka vai komanda ir starp vadošajām komandām līgā. Algoritms pārbauda sekojošu komandas sezonas statistiku: kopvērtējuma punkti, uzbrukuma efektivitāte, aizsardzības efektivitāte.

Katrs algoritms, ja konstatē interesantu faktu izveido teikumu, kas vislabāk raksturo konstatēto faktu. Katrs teikums ir iekļauts vienā no sešām grupām: Mājinieku komandas stiprās puses, mājinieku komandas vājās puses, viesu komandas stiprās puses, viesu komandas vājās puses, savstarpējās spēlēs pozitīvie notikumi mājinieku komandai, savstarpējās spēlēs pozitīvie notikumi viesu komandai. Teikumu grupēšana ļauj vieglāk orientēties apskatā un atrast vēlamu informāciju par komandām. Balstoties uz izveidotajiem algoritmiem tika ģenerēti spēļu apraksti visām spēlēm sākot ar 2013. gada oktobri. Iegūtie rezultāti ir ļoti dažādi, dažām spēlēm ir plašs apskats dažām spēlēm atrasto faktu ir pavisam maz, dažām spēlēm apskatu neizdodas uzģenerēt, iespējams kādu specifiku statistikas neatbilstību dēļ. Plašāki raksti parasti ir līdervienībām vai vājākajām komandām, jo šo komandu statistika stabilāka un vairāk atšķiras no līgas vidusmēra.

Tekstu ģenerēšanai tiek izmantota latviešu valoda. Tomēr latviešu valodas atbalstu pilnīgu ieviest neizdevās, jo komandu nosaukumi un spēlētāju vārdi ir palikuši angļu valodā. Komandām nosaukumus pārtulkot varētu būt samēra vienkārši, bet spēlētājiem vārdus pat profesionāli žurnālisti nespēj korekti iztulkot, tāpēc nemaz netika mēģināts to darīt automātiski. Izvēlētā teksta ģenerēšanas struktūra, kas balstās uz veidņu aizpildīšanu, dod iespēju ērti pievienot vēl valodas. Izvēlētajam risinājumam būtībā vienlīdz kāda valoda tiktu izmantota, jo veidnes varētu pielāgot jebkurai man zināmai dabīgajai valodai. Teksta konvertēšana audio formātā citās valodās varētu būt būtiski vieglāka nekā latviešu valodā. Vispiemērotākā valoda šādam uzdevumam būtu angļu valoda.

Turpinot sistēmas attīstību izmantojot jau esošās iestrādes būtu iespējams izveidot sistēmu, kas varētu attēlot pietiekami daudz interesantu statistikas faktu un varētu būt kā vērtīgs resurss komandu spēju novērtēšanai pirms spēles.

## SECINĀJUMI

Nebiju līdz šim domājis, ka tik liela daļa rakstu mūsdienās tiek automātiski ģenerēti, kaut gan ikdienu izmantoju epastu, kurā teju visa saņemtā informācija ir datora ģenerēta. Automātisku raksti ir daudz, bet sistēmas, kas spētu ģenerēt komplicētus kontekstam atbilstošus rakstus, kuros teikumi tiek konstruēti un savstarpēji saistīti ir retums. Parasti automātiski ģenerētie raksti balstās uz stingrām veidnēm kurās izmainās tikai daļa teksta.

Dabīgās valodas ģenerēšanas sistēmai nav jābūt sarežģītai, lai būtu lietderīga. Pirmajā nodaļā tika minēta zemestrīču uzraudzības sistēma, kas veido rakstus žurnālam “LA Times” par notikušajām zemestrīcēm. Šādas mazas sistēmas ir spējīgas ātri veidot īsus saturīgus aprastus iekļaujot visu nepieciešamo informāciju. Mākslīgajam intelektam nav jāmaksā mēnešalga un nav noteikts darba laiks, tāpēc ir pavisam skaidrs, ka līdzīgas sistēmas žurnālistikā un citās nozarēs paliks tikai arvien vairāk. Manuprāt, pēc aptuveni 10 gadiem par normu uzskatīs amatieru sporta līgām automātiski ģenerētus rakstus, kas ļoti līdzināsies profesionālo līgu rakstiem.

Izveidot sistēmu, kas spēj atrast interesantus statistikas faktus par hokeja komandām ir samērā viegli, bet apvienot šos faktus un izveidot stāstījumu ir būtiski grūtāk. Sarežģītība starp šādiem uzdevumiem pieaug tik strauji, ka veltītais laiks izveidojot sistēmu, kas spētu izveidot stāstījumu varētu pārāk liels, lai tam veltītu laiku, jo cilvēks spēj labi uztvert kopējo domu kaut apskats ir veidots tikai no atsevišķiem faktiem.

Mākslīgā intelekts, kas spētu saprast un izmantot dabīgās valodas cilvēka līmenī ir tālā nākotnē, jo izveidot sistēmu, kas veido dabīgās valodas tekstus, jau ir ļoti komplicēts process un tā attīstīšana prasīs daudz laika un resursu, bet datoram spēt pareizi interpretēt cilvēku valodā rakstītos tekstus ir vēl pakāpi sarežģītāk par tekstu radīšanu.

## LITERATŪRAS SARAKSTS

1. Kris Hammond blog [tiešsaiste]. [skatīts 27.05.2015.]. Pieejams: <http://khammond.blogspot.com/2011/12/why-90-of-news-will-be-computer.html>
2. NHL official homepage [tiešsaiste]. [skatīts 27.05.2015.]. Pieejams: <http://www.nhl.com/ice/schedulebyseason.htm?season=20142015&gameType=2&team=&network=&venue=>
3. Story Generator Algorithms [10. paragrāfs] [tiešsaiste]. [skatīts 27.05.2015.]. Pieejams: [http://wikis.sub.uni-hamburg.de/lhn/index.php/Story\\_Generator\\_Algorithms](http://wikis.sub.uni-hamburg.de/lhn/index.php/Story_Generator_Algorithms)
4. Choosing Words in Computer-Generated Weather Forecasts [tiešsaiste]. [skatīts 27.05.2015.]. Pieejams: <http://www.csd.abdn.ac.uk/~ssripada/sumtime/aij05.pdf>
5. SUMTIME-MOUSAM: Configurable Marine Weather Forecast Generator [tiešsaiste]. [skatīts 27.05.2015.]. Pieejams: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.8903>
6. Earthquake aftershock: 2.7 quake strikes near Westwood [tiešsaiste]. [skatīts 27.05.2015.]. Pieejams: <http://www.latimes.com/local/lanow/earthquake-27-quake-strikes-near-westwood-california-rdivor-story.html#ixzz2wEWmfUmX>
7. The journalists who never sleep [tiešsaiste]. [skatīts 27.05.2015.]. Pieejams: <http://www.theguardian.com/technology/2014/sep/12/artificial-intelligence-data-journalism-media>
8. Robot writes LA Times earthquake breaking news article [tiešsaiste]. [skatīts 27.05.2015.]. Pieejams: <http://www.bbc.com/news/technology-26614051>
9. Natural Language Generation: An Overview [tiešsaiste]. [skatīts 30.05.2015.]. Pieejams: <http://www.lacsc.org/papers/PaperA6.pdf>
10. The Architecture of a Natural Language Generation System [tiešsaiste]. [skatīts 30.05.2015.]. Pieejams: <http://www.ling.helsinki.fi/~gwilcock/Tartu-2003/ReiterDale/3-Architecture.pdf>
11. Data-Driven Communications at Machine Scale [tiešsaiste]. [skatīts 27.05.2015.]. Pieejams: <http://www.narrativescience.com/quill>

12. Open Source Sports [tiešsaiste]. [skatīts 27.05.2015.]. Pieejams: <http://www.opensourcesports.com/hockey/>
13. DB-Engines Ranking [tiešsaiste]. [skatīts 27.05.2015.]. Pieejams: <http://db-engines.com/en/ranking>
14. Ranking Programming Languages [tiešsaiste]. [skatīts 31.05.2015.]. Pieejams: <http://readwrite.com/2015/01/15/popular-programming-languages-swift-go-r-redmonk>
15. Laravel documentation [tiešsaiste]. [skatīts 31.05.2015.]. Pieejams: <http://laravel.com/docs/master/quickstart>
16. Google Text-to-Speech [tiešsaiste]. [skatīts 30.05.2015.]. Pieejams: [http://translate.google.com/translate\\_tts?tl=lv&q=testa%20runa](http://translate.google.com/translate_tts?tl=lv&q=testa%20runa)
17. Ivona Text-to-Speech [tiešsaiste]. [skatīts 30.05.2015.]. Pieejams: <http://www.ivona.com/>
18. Audio Processing for Automatic TV Sports Program Highlights Detection [tiešsaiste]. [skatīts 30.05.2015.]. Pieejams: [http://doras.dcu.ie/326/1/issc\\_2002\\_2.pdf](http://doras.dcu.ie/326/1/issc_2002_2.pdf)
19. A HtmlDom package for Laravel 4 based on Simple HTML Dom Parser [tiešsaiste]. [skatīts 30.05.2015.]. Pieejams: <https://github.com/yangqi/HtmlDom>

1. pielikums **Pirmsspēles apskata paraugs (kā vienkāršs teksts) attēlotā spēle:**<http://www.nhl.com/gamecenter/en/preview?id=2014021151>

Vispārīgi fakti

Anaheim

Stiprās puses

Spēlējusi biežāk vairākumā 3 spēles pēc kārtas

Ievērtības cienīgu sniegumu pdejās spēlēs rādījis C. Perry, pēdējās 4 spēlēs gūstot 1 vārtus un izdarot 1 piespēles

Ievērtības cienīgu sniegumu pdejās spēlēs rādījis R. Getzlaf, pēdējās 4 spēlēs gūstot 2 vārtus un izdarot 2 piespēles

Komanda ieņem augsto 1. vietu līgā

Nopelnījuši mazāk soda minūtes 3 spēles pēc kārtas

Izdarījusi vairāk metienus 2. periodā 3 spēles pēc kārtas

Izdarījusi vairāk metienus 2. periodā 7 no 8 pēdējām spēlēm

Uzvarējusi 3 spēles pēc kārtas

Vājās puses

Zaudējusi iemetienos 3 spēles pēc kārtas

Zaudējusi iemetienos 5 no 6 pēdējām spēlēm

Mazāk piespiesto kļūdu kā pretiniekam 5 no 6 pēdējām spēlēm

Izdarījusi mazāk metienus 1. periodā 7 no 8 pēdējām spēlēm

Izdarījusi mazāk metienus 3. periodā 5 no 6 pēdējām spēlēm

Edmonton

Stiprās puses

Spēlējusi biežāk vairākumā 3 spēles pēc kārtas

Ievērtības cienīgu sniegumu pdejās spēlēs rādījis A. Miller, pēdējās 4 spēlēs gūstot 1 vārtus un izdarot 5 piespēles

Ievērtības cienīgu sniegumu pdejās spēlēs rādījis T. Hall, pēdējās 3 spēlēs gūstot 1 vārtus un izdarot 3 piespēles

Uzvarējusi iemetienos 5 no 6 pēdējām spēlēm

Pieļāvusi mazāk nepiespiestās kļūdas 6 no 7 pēdējām spēlēm

Nopelnījuši mazāk soda minūtes 3 spēles pēc kārtas

Izdarījusi vairāk metienus 2. periodā 3 spēles pēc kārtas

Uzvarējusi 3 spēles pēc kārtas

Uzvarējusi 5 no 6 pēdējām spēlēm

Vājās puses

Komanda ieņem zemo 28. vietu līgā

Komandai ir 5. vājākais uzbrukums līgā

Komandai ir 1. vājākā aizsardzība līgā

Mazāk piespiesto kļūdu kā pretiniekam 5 spēles pēc kārtas

Mazāk piespiesto kļūdu kā pretiniekam 6 no 7 pēdējām spēlēm

Izdarījusi mazāk metienus 3. periodā 4 spēles pēc kārtas

Izdarījusi mazāk metienus 3. periodā 6 no 7 pēdējām spēlēm

Izdarījusi mazāk metienus 5 no 6 pēdējām spēlēm

Savstarpējo spēļu fakti

Anaheim

Izdarījusi vairāk spēka paņēmieni 3 spēles pēc kārtas

Izdarījusi vairāk metienus 3. periodā 3 spēles pēc kārtas

Uzvarējusi 3 spēles pēc kārtas

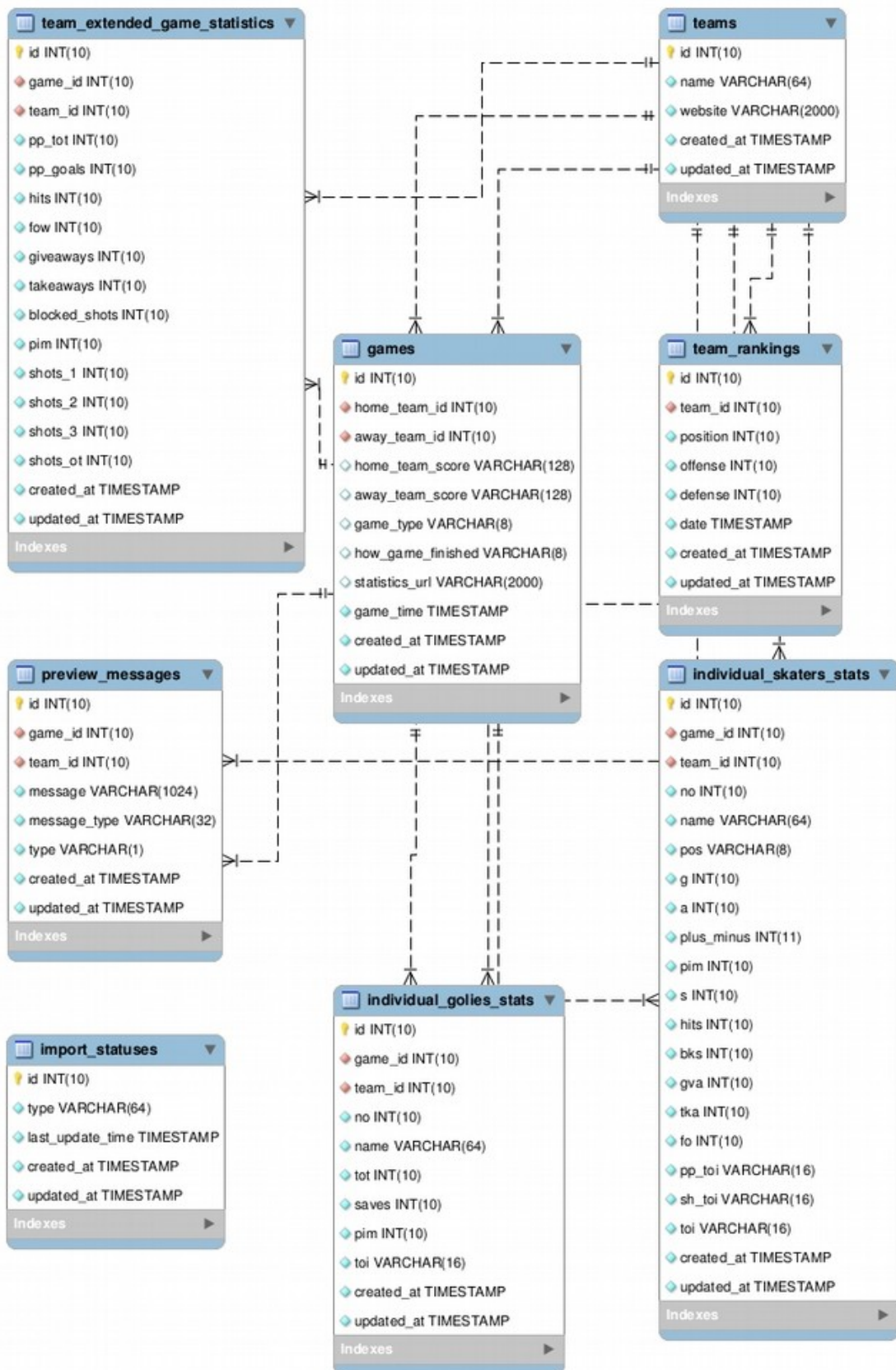
Uzvarējusi 5 no 6 pēdējām spēlēm

Edmonton

Izdarījusi vairāk metienus 3 spēles pēc kārtas

Izdarījusi vairāk metienus 2. periodā 5 no 6 pēdējām spēlēm

## 2. pielikums Sistēmas datubāzes shēma



### 3. pielikums Sistēmas uzstādīšanas pamācība

1. Instalēt nginx hhvm composer

<http://fideloper.com/hhvm-nginx-laravel>

2. Instalēt mysql

<https://help.ubuntu.com/lts/serverguide/mysql.html>

3. mkdir for project

```
sudo mkdir /www
```

```
sudo chown kazha:users /www/
```

4. Noklonēt projektu no git repozitorija

```
cd /www/
```

```
git clone ssh://current_local_ip/home/kazha/Dropbox/Git/sports_predictor/
```

5. Izmantojot composer update komandu lejuplādēt visas nepieciešamās laravel bibliotēkas kas netika glabātas git

```
cd /www/sports_predictor
```

```
composer update
```

```
** ja nepieciešams export LC_ALL=C
```

6. Piešķirt projektam failiem pareizas pieejas tiesības un nokonfigurēt nginx

```
sudo chmod -R guo+w /www/sports_predictor
```

```
sudo vim /etc/nginx/sites-available/sports_predictor
```

7. Izveidot mysql lietotāju un datubāzi

<https://www.digitalocean.com/community/tutorials/how-to-create-a-new-user-and-grant-permissions-in-mysql>

```
CREATE DATABASE HockeyStats;
```

```
CREATE USER 'hockey'@'localhost' IDENTIFIED BY 'predictor_hockey';
```

```
GRANT ALL PRIVILEGES ON HockeyStats.* TO 'hockey'@'localhost';
```

```
php artisan migrate
```

8. Peivienot crontab uzdevumu

```
* * * * * php /www/sports_predictor/artisan schedule:run 1>> /dev/null 2>&1
```

veikt pagaidu izmaiņas lai ātrāk importētu datus /www/sports\_predictor/app/Console/Kernel.php

Bakalaura darbs „Hokeja pirmsspēles apskata automātiska veidošana” izstrādāts Latvijas  
Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie  
informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Kārlis Freimanis

06.01.2015.

Rekomendēju/nerekomendēju darbu aizstāvēšanai

Darba vadītājs: profesors Dr.dat. Guntis Arnicāns

06.01.2015.

Recenzents: profesore Dr.dat. Laila Niedrīte

Darbs iesniegts Datorikas fakultātē 01.06.2015.

Dekāna pilnvarotā persona: metodiķe Ārija Sproģe

Darbs aizstāvēts bakalaura darbu pārbaudījuma komisijas sēdē

\_\_\_\_.\_\_\_\_.2015. prot. Nr. \_\_\_\_\_

Komisijas sekretārs(-e): \_\_\_\_\_