Rīgas 64. Vidusskola

**Kāda Konkrēta YouTube Video Komentāru Sadaļas Sentimenta Nolasīšana: *VADER* un *RoBERTa* Modeļu Salīdzinājums**

Zinātniski pētnieciskais darbs: Datorzinātņu un datorlingvistikas sadaļā

**Darba autors:**

Rīgas 64. vidusskolas 12. klases skolnieks

Kristers Laganovskis

**Darba vadītājs:**

Rīgas 64. vidusskolas programmēšanas skolotājs

Edvards Bukovskis

Rīga 2023

**Anotācija**

Zinātniski pētniecisko darbu: “Sentimenta novērtēšana YouTube komentāru sadaļai” Izstrādājis autors Rīgas 64. vidusskolas 12.DIT klases skolēns – Kristers Laganovskis. Darba Vadītājs Rīgas 64. vidusskolas programmēšanas skolotājs - Edvards Bukovskis.

Zinātniski pētnieciskais darbs tika veikts ar mērķi, izpētīt, kā ar dabiskās valodas apstrādi (NLP) iegūst sentimenta analīzi. Kādas ir būtiskākās atšķirības starp *VADER* un *RoBERTa* modeļiem. Kā, tiek veikta efektīva datu ievākšana. Kā, tiek apstrādāti dati. Kā arī, cik precīzi ir iespējams novērtēt komentāru sadaļu balsoties uz mākslīgo intelektu un programmēšanas palīdzību. Zinātniski pētnieciskajā darbā balstoties uz analizēt un aplūkoto, tika izveidots pārlūka paplašinājums, ar kura palīdzību iespējams analizēt sentimentu kādam konkrētam YouTube video komentāru sadaļai.

Teorētiskajā daļa tika detalizēti izpētīti dabiskās valodas apstrādes (NLP) pamati, teksta sentimenta analizēšanai. Tās pielietošana sociālajā un ētiskajā kontekstā, kā arī tās nozīme mūsdienu un vēsturiskajā kontekstā. Tika apskatīts, kas ir *VADER* modelis un kas ir *RoBERTa* modelis un to atšķirības. Kā, tiek veikta efektīva datu ievākšana un tā principi. Kā arī datu apstrādes principi, lai nodrošinātu pēc iespējas precīzāku sentimenta novērtējumu. Kā arī tika apsvērti sentimenta analīzes ierobežojumi.

Praktiskā daļā tika izstrādāts pārlūka paplašinājums, kas veidots uz VEDER un *RoBERTa* modeļu bāzes, šis paplašinājums sniedz iespēju vizualizēt sentimentu statistiski par kāda konkrēta YouTube video komentāru sadaļu, sniedzot vērtīgu perspektīvu par to, kāda ir publikas reakcija uz konkrēto saturu video. Paplašinājums nodrošina iespēju efektīvi filtrēt komentārus pēc to sentimenta, padarot analīzes procesu vēl efektīvāku. Taču šāda veida analīzei kļūdas ir tik lielas, ka visticamāk tam ir neliela praktiska pielietojuma vērtība. Iespējamie metodes uzlabojumi ietver nevēlamo reklāmu komentāru filtrēšanu un “emoji” sentimenta iekļaušanu.

Atslēgas Vārdi: Datu ievākšana, datu apstrāde, Komentāri, sentimenta analīze, pārlūka paplašinājums, dabiskās valodas apstrāde (NLP), mašīnmācīšanās, *VADER*, *RoBERTa*.

**Abstract**

Scientific Research Work: "Sentiment Analysis of YouTube Comment Section" Developed by the author, a student of the 12th-grade DIT class at Riga Secondary School No. 64 – Kristers Laganovskis. Supervisor: Edvards Bukovskis, programming teacher at Riga Secondary School No. 64.

The scientific research work was conducted with the aim of exploring how sentiment analysis is obtained through Natural Language Processing (NLP). It delves into the significant differences between the *VADER* and *RoBERTa* models, as well as how to efficiently gather data and process it. Additionally, it evaluates the precision of assessing the comment section using artificial intelligence and programming assistance. Based on the analysis and examination of the scientific research work, a browser extension was developed to analyze the sentiment of specific YouTube video comment sections.

In the theoretical part, the fundamentals of Natural Language Processing (NLP) for text sentiment analysis were thoroughly studied, including its application in social and ethical contexts, as well as its significance in modern and historical contexts. The *VADER* model and the *RoBERTa* model, along with their differences, were examined. The effective principles of data collection and processing were also discussed to ensure the most accurate sentiment assessment possible. The limitations of sentiment analysis were also considered.

In the practical part, a browser extension was developed based on the *VADER* and *RoBERTa* models. This extension provides the opportunity to statistically visualize the sentiment of a specific YouTube video comment section, offering valuable insight into the audience's reaction to the specific content of the video. The extension allows for the efficient filtering of comments by their sentiment, making the analysis process even more effective. However, the errors in this type of analysis are likely so significant that it has little practical application value. Possible method improvements include filtering out unwanted advertising comments and incorporating “emoji” sentiment.

Keywords: Data collection, data processing, comments, sentiment analysis, browser extension, Natural Language Processing (NLP), machine learning, *VADER*, *RoBERTa*.

**Saturs**

**Anotācija**…......................................................................................................................................2

**Abstract**...........................................................................................................................................3

**Ievads**…………………………………………………………………………………………………………………………………..…..5

**1. Literatūras apskats**…………………………………………………………………..…………………..……………………….6

1.1 Dabiskās Valodas Apstrādes (NLP) Pamati………………………………………………………………….………..6

1.2 Sentimenta nolasīšanas principi (NLP)…………………………………………………………………………………7

1.2.1 Datu iegūšana……………………………………………………………………………………………………………7

1.2.2 Datu apstrāde……………………………………………………………………………………………………………7

1.2.2.1 Datu apstrādes metodes……………………………………………………………………………….8,9

1.2.3 Sentimenta nolasīšana………………………………………………………………………………………….9,10

1.3 *VADER* un *RoBERTa* modeļu salīdzināšana………………………………………………………………………….10

1.4 Datu ievākšana no YouTube video komentāru sadaļas………………………………………………

**Secinājumi**………………………………………………………………………………………………………………….……………..0

**Izmantotie informācijas avoti**…………………………………………………………………………………………………….0

**Pielikumi**……………………………………………………………………………………………………………………………………0

**Ievads**

Mūsdienās informācijas plūsma ir pārāk bagātīga, un dažkārt tas var radīt problēmu gan saturu patērētājam, gan to veidotājiem. Būtisks šķērslis ir tas, ka cilvēki bieži vien vērš pārāk lielu uzticību satura veidotājiem, kas, kā rāda pieredze, var nebūt vienmēr objektīvi. Šāda pieeja rada risku, ka satura veidotājs varētu apslēpt vai pat manipulēt ar informāciju, ietekmējot patērētājus. Tādēļ ir ārkārtīgi svarīgi uzzināt, kā citi cilvēki novērtē konkrētā video saturu. Īpaši šo var novērot interneta platformā "YouTube", kur satura veidotāji bieži veicina savu subjektīvo viedokli, ko patērētāji bieži uztver kā objektīvu faktisko informāciju. Tomēr tas var radīt lielas problēmas, jo ne vienmēr šim veidotājam būs pareiza nostāja. Tāpēc ir svarīgi uzzināt, ko citi cilvēki domā par konkrēto viedokli, vai saturu, un veikt pašam savus secinājumus. Veikt secinājumus, pārskatot pāris komentārus, ir ātri un ērti, bet, kad to ir simtiem vai tūkstošiem, šāda analīze prasa ievērojami vairāk laika. Tāpēc sentimenta analīze ir lietderīga, jo tā ļauj strukturēt komentāru sadaļu nostājās - pozitīvas, neitrālas vai negatīvas.

Šajā pētījumā tiks apskatīts, kāda YouTube video komentāru sadaļa, izmantojot valodas apstrādes (NLP) modeļus, spēj nolasīt komentāru noskaņojumu, izmantojot *VADER* un *RoBERTa* modeļus. Tiks analizētas metodes, kas tiek izmantotas, lai iegūtu nepieciešamos datus analīzei, kā arī kā šie dati ir jāapstrādā, lai sasniegtu veiksmīgu sentimenta novērtējumu. Turklāt tiks izvērtēta mākslīgā intelekta spēja precīzi noteikt komentāru noskaņojumu. Papildus tam, tiks izveidots paplašinājums, kas ļaus ikvienam analizēt komentārus un tos filtrēt pēc noteiktām kategorijām. Tas būs noderīgs ne tikai satura patērētājiem, bet arī veidotājiem, ļaujot tiem ietaupīt laiku, filtrējot komentārus.

**Darba mērķis:** Izpētīt dabiskās valodas (NLP) principus kā arī salīdzināt *VADER* un *RoBERTa* modeļus un izveidot paplašinājumu, kas ļaus analizēt komentārus.

**Hipotēze:** Autoram izdosies veiksmīgi salīdzināt *VADER* un *RoBERTa* modeļus kā arī analizēt un interpretēt YouTube video komentārus, izmantojot (NLP) modeļus un programmēšanu.

**Darba uzdevumi:**

1. Izpētīt (NLP) modeli un kā to pielietot sentimenta novērtēšanai.
2. Izpētīt atšķirības starp *VADER* un *RoBERTa* modeļiem.
3. Izpētīt, kā iegūt nepieciešamos datus no YouTube video komentāru sadaļas.
4. Izpētīt, kā apstrādāt datus efektīvi, lai tie sniegtu precīzāku novērtējumu.
5. Izstrādāt paplašinājumu, kas ļauj ikvienam analizēt un filtrēt komentārus pēc noteiktām kategorijām.
6. Veikt secinājumus

**Izmantotās darba metodes:** Literatūras apskats, lai varētu izpētīt, Dabiskās valodas apstrādes (NLP). Salīdzināšanas metode, lai varētu salīdzināt *VADER* un *RoBERTa* (NLP) modeļus. Sentimenta nolasīšanas metode, lai interpretētu iegūtos datus. Statistikas un analīzes metodes. Atvērtā pirmkoda pielietošana, “Python” programmēšanas valodas pielietošana radošās daļas veikšanai.

**Darba struktūra:** Darbs sastāv no x nodaļām, x apakšnodaļām, secinājumiem, izmantoto literatūras avotu saraksta un x pielikumiem. Darbā ir x attēli.

1. **Literatūras apskats**
   1. **Dabiskās Valodas Apstrādes (NLP) Pamati**

Dabiskās Valodas Apstrāde (angliski: natural language processing, turpmāk — NLP) ir mākslīgā intelekta joma, kas nodarbojas ar datoru spēju saprast, analizēt un interpretēt cilvēku valodu. Tās sākotnējie pētījumi sākās jau 1950. un 1960. gadu mijā, kad tika veidoti pirmie mēģinājumi izveidot datorprogrammas, kas spētu analizēt un interpretēt tekstu. Šajā laikā tika izveidots pirmās valodas analīzes algoritms, kas varēja identificēt vārdus, frāzes un sintaktiskos elementus. [1] Kopš tā laika NLP ir piedzīvojis ievērojamu attīstību un progresu, pateicoties tehnoloģiju un algoritmu uzlabojumiem. Ar NLP mūsdienu pasaulē mēs saskaramies teju katru dienu, jo to pielieto plaši ikdienas ērtībās ko ikkatrs lietojam. To izmanto balss vadības GPS sistēmas, digitālajos asistentos, runas-uz-tekstu dikcijas programmās, tulkošanas programmās, klientu apkalpošanas “čatbotos” un citās patērētāju ērtībās. [2]

Tomēr NLP arvien lielāku lomu ieņem uzņēmējdarbības risinājumos, kas palīdz optimizēt uzņēmējdarbības darbības, palielina darbinieku produktivitāti un vienkāršo būtiskus uzņēmējdarbības procesus. [2] Arvien biežāk NLP tiek izmantota arī medicīnas jomā, lai analizētu medicīnisko dokumentāciju un izgūtu noderīgu informāciju ārstiem un pētniekiem. [3]

Sentimenta nolasīšana, jeb analīze ir viena no svarīgākajām NLP apakšnozarēm. Tā ļauj noteikt teksta vai runas emocionālo noskaņu, palīdzot izsekot un analizēt viedokļus un emocijas. Sentimenta nolasīšana piedāvā plašas pielietošanas iespējas, tostarp sociālajos medijos, uzņēmējdarbībā, mārketingā, pētniecībā un citviet. [4]

Sentimenta analīze tiek veikta, izmantojot dator algoritmus, kas analizē vārdus, frāzes, un teikumus, lai noteiktu to emocionālo saturu. Šie algoritmi izmanto vārdnīcas un emociju leksikonus, lai klasificētu vārdus un frāzes polaritātes: pozitīvs, negatīvs vai neitrāls. Tas ļauj kopēji novērtēt teksta noskaņu un viedokļus. [4]

Sentimenta analīze ir īpaši noderīga, lai novērtētu un klasificētu lielus teksta apjomus, piemēram, sociālo mediju ierakstus, produktu atsauksmes vai klientu aptaujas. Tā palīdz uzņēmumiem iegūt ieskatu par to, kā viņu produkti vai pakalpojumi tiek uztverti tirgū, un sniedz iespēju uzlabot darbību, balstoties uz klientu atsauksmēm un viedokļiem. [5]

NLP ir kļuvusi par neatņemamu mūsdienu tehnoloģiju daļu, kas iesaistīta gan ikdienas patērētājus, gan uzņēmējdarbības risinājumus. Tā piedāvā iespēju saprast un analizēt cilvēku valodu, atvieglojot daudzas ikdienas darbības un uzlabojot uzņēmējdarbības procesus. Svarīga apakšnozare - sentimenta nolasīšana - kas palīdz novērtēt teksta emocionālo saturu un piedāvā plašas pielietošanas iespējas dažādās nozarēs. Tas veicina uzņēmumu izaugsmi, sniedzot iespēju labot un pielāgot piedāvātos produktus un pakalpojumus, balstoties uz klientu atsauksmēm un viedokļiem.

**1.2 Sentimenta nolasīšanas principi**

Kā jau iepriekš tika minēts, sentimenta nolasīšana ir viena no svarīgākajām NLP apakšnozarēm. Tāpēc šajā sadaļā tiks dziļāk apskatītas sentimenta analīzes pamatprincipi un tehnikas. Sentimenta nolasīšanas mērķis ir saprast un klasificēt tekstu vai runu, lai noteiktu tā emocionālo noskaņojumu. Sentimentu var klasificēt pēc polaritātes – pozitīvi, neitrāli un negatīvi, vai pēc vajadzības vēl mazākos iedalījumos. Sentimenta nolasīšanai, ir jāveic trīs būtiski soļi. Pirmais solis ir iegūt datus, otrais ir apstrādāt datus un trešais ir veikt datu analīzi jeb sentimenta nolasīšanu. [6]

**1.2.1 Datu ievākšana**

Šis ir pirmais solis sentimenta nolasīšanā. Tas ietver iegūto datu kopumu, kuram tiks veikta sentimenta analīze. Šie datus var iegūt no ziņām, komentāriem, sociālo mediju ierakstiem, klientu atsauksmes, klientu apkalpošanas centra datiem, darbinieku mijiedarbības datiem un citu veidu datiem, lai veiktu sentimenta nolasīšanu no tiem. [7] Datu ievākšanu var veikt ar dažādām metodēm. Viena no populārākajiem veidiem ir lietot *API (Lietojumprogrammas saskarne)*, ko nodrošina sociālo mediju platformas, kas ļauj savākt, datus straumējot. Piemēram: *Twitter* *API*, lai iegūtu ziņas ar noteiktām “heštegu” atsauksmēm, *News API*, lai izgūtu ziņas pēc kategorijām no dažādiem ziņu izdevējiem. Otrs populārs veids ir izmanot *web scraping (rasmošana)*, kas pārmeklē tīmekļa datus un savāc norādīto informāciju. Piemēram var izmantot *Python* bibliotēku *BeautifulSoup*, lai iegūtu jebkādu informāciju no tīmekļa, piemēram, ziņu raksti vai komentāri no blogiem, analizējot *HTML* tagus. Datus var ievākt arī izmantojot tīmekļa pārlūka paplašinājumus, ar kura palīdzību lietotājs var izgūt informāciju no jeb kuras publiskas tīmekļa vietnes un tad eksportējot datus vēlamā faila formātā. Piemēram: *Webscraper.io*. Kā arī datus var, ievākt no esošiem datu krājumiem, kas jau ir apstrādāti un ir gatavi lietošanai. Piemēram: *Rotten Tomatoes*, *IMDB* filmu apskats, *Yelp*, *Amazon* produkta apskats, *Twitter* tvīti un citas. [8]

**1.2.2 Datu apstrāde**

Datu apstrāde: Otrais svarīgais solis sentimenta noteikšanā ir iegūto datu apstrāde. Datu apstrāde ir kritisks solis sentimenta nolasīšanā, jo labi apstrādāti dati, veido pamatu efektīvai sentimenta klasifikācijas modeļa izveidei. Teksta apstrāde ir process, kurā tiek iepriekš iegūtie dati attīrīti un apstrādāti. Teksta jeb datu apstrādē ietilpst procesi kā - pieturzīmju izņemšana, bieži sastopamu vārdu noņemšana *(stop word removal),* visu tekstu pārveidojot, mazajos burtos *(lowercasing)*, lemmatizācija, tokenizācija *(tokenization)*, noliegumu apstrāde, intensitāšu apstrāde, *emojie* apstrāde, vektorizācija *(vectorization)* un retu vai zemu frekvenci vārdu apstrāde. [9] Datu zinātniekiem ir teiciens *“garbage in, garbage out”*, kas tulkojas kā *“atkritumi iekšā, atkritumi ārā.”* Šī teiciena nozīme ir uzsvērt to, ka datu apstrāde ir ļoti svarīgs process sentimenta noteikšanā, jo gala rezultāts būs atkarīgs no tā cik kvalitatīvi dati tiks padoti analīzes modelim. [10]

**1.2.2.1 Datu apstrādes metodes**

Kā, jau autors minēja iepriekš ir daudz un dažādas metodes, kā apstrādāt tekstu jeb datus. Taču, katra metode pilda noteiktu funkciju datu apstrādes procesā un ne vienmēr būs būtiski izmantot visas metodes kopā. Tāpēc tagad tiks izskatītas, pāris galvenās un nozīmīgākās metodes atsevišķi.

1. **Metode visa teksta pārveidošana mazajos burtos** ***(lowercasing)***: Viss teksts tiek pārveidots mazajos burtos, lai nodrošinātu vienmērību un novērstu iespējamu vārdu dublēšanos ar atšķirīgiem lielajiem un mazajiem burtiem. Piemēram: vārdi “Labs” un “labs” tiks uzskatīti par vienu un to pašu vārdu. [9] Šo metodi izmanto, lai ietaupītu resursus, jo tas samazina vārdu skaitu, kas ir jāapstrādā. Kā arī strādāt, ar viena formāta vārdiem ir vieglāk, tas palīdz uzlabot precizitāti un vienkāršo algoritma darbu. Šo metodi izmanto, ja ir daudz vārdu, kas dublējas, taču nav vienā formātā un šo vārdu nozīme nemainās, kad tiek izmantoti lielie burti, piemēram, sociālajos mēdijos. Šo metodi var veikt ar *Python* programmēšanas valodu.[11]
2. **Metode tokenizācija *(tokenization)*:** Viss teksts tiek sadalīts individuālos vārdos, kurus sauc par *tokeniem*, šis ir būtisks process datu apstrādē, jo tas ļauj standartizēt ievades datus modelī, padarot tos vieglāk apstrādājamus, lai analizētu teksta sentimentu. Jo bez tokenizācijas *(tokenization)* metodes sentimenta noteikšanas modelim būtu jāstrādā ar visu tekstu kā vienu virkni ar burtiem, kas prasītu lielus resursus un padarītu datus sarežģīti modelējamus. Šo metodi var veikt ar *Python* bibliotēkas *nltk* palīdzību. [12] Piemēram: Teikums no ievāktajiem datiem “Šodien ir saulaina diena.”, tiktu pārveidots pēc tokenizācijas *(tokenization)* uz “[‘Šodien’, ‘ir’, ‘saulaina’, ‘diena’, ‘.’]”, taču, ja nebūtu piemērota toeknizācija *(tokenization)* un tiktu padoti ievades dati modelim bez tās, tad teksts izskatītos šādi - “Šodienirsaulinadiena.” Šo metodi izmanto vienmēr, kad veic datu apstrādi sentimenta noteikšanai.
3. **Metode atbrīvošanās no visām pieturzīmēm:** No visa teksta tiek izņemtas ārā visas pieturzīmes un tagi. Tas iekļauj – komatus, punktus, izsaukuma zīmes un citas zīmes. Pieturzīmes nenes lielu informācijas vērtību sentimenta nolasīšanas modelim un tās var tikt noņemtas, lai samazinātu tekstā troksni. [11]
4. **Metode bieži sastopamu vārdu noņemšana *(stop word removal)*:** Bieži sastopami vārdi, piemēram, "un", "bet", "ir", utt. Tiek noņemti, jo tie nenes lielu nozīmi, lai noteiktu sentimentu un pat var negatīvi ietekmēt sentimenta rezultātu. Šo metodi izmanto vienmēr, kad veic datu apstrādi sentimenta noteikšanai. [11]
5. **Metode lemmatizācija un pamatvārda noteikšana *(Lemmatization and Stemming)*:** Lemmatizācija attiecas uz vārda pamatformas noteikšanu (parasti tas ir vārds, ko var atrast vārdnīcā). Piemēram, vārda 'ābols' lema joprojām būtu 'ābols', bet vārda 'ir' lema būtu 'būt'. Lemmatizācija ir pamats katrai sentimenta nolasīšanas procesam. Tā ņem vērā dažādās esošās valodas struktūras, un katrā valodā lematizācija var būt atšķirīga. [13] Pamatvārda noteikšana ir process, kurā vārdi tiek samazināti līdz to pamatformai, atgriežot tos vispārīgāko veidā.

Piemēram, vārda "mājas" pamatvārds būtu "māj", un vārda "gāju" pamatvārds būtu "gāj". Pamatvārda noteikšana ir vienkāršāka un mazāk resursu prasīga nekā lematizācija, bet var būt mazāk precīza. Galvenais atšķirības punkts ir tas, ka lematizācija cenšas saglabāt vārda nozīmi, samazinot to līdz pamatformai, savukārt pamatvārda noteikšana vienkārši mēģina atgriezt vārdu vispārīgākajā formā. [14]

1. **Metode negāciju apstrāde un intensifikatori *(handaling negations and intemsifiers)*:** Noliegumu apstrāde ir metode, kas nosaka nolieguma ietekmes apjomu uz viedokļa vārdiem, ko ietekmē noliegums. Noliegumi ir vārdi kā 'nav', 'ne', 'nekad', utt., kas ietekmē citu vārdu noskaņu frāzē. Noliegums var pilnībā mainīt citu vārdu noskaņu vienvirziena teikumā. Ir trīs noliegumu kategorijas: sintaktiskie, samazinātāji un morfoloģiskie noliegumi. [15] Intensifikatori *(intemsifiers),* piemēram “ļoti” vai “ārkārtīgi”, maina vārda nozīmi. Intensifikatori *(intemsifiers)* ir apstākļa vārdi, kas pastiprina citu izteicienu nozīmi. Par pastiprinātājiem parasti tiek lietoti tādi vārdi kā absolūti, pilnīgi, ārkārtīgi, ļoti. Sentimenta analīzē noliegumu un intensifikatoru apstrāde ir būtiska, jo tā var mainīt teikuma noskaņu. *(skatīt 1. pielikumā)*

Šis ir tās metodes, kuras ir jāņem vērā, apstrādājot iegūtos datus. Lai gan, protams, ir daudzas citas metodes, kas īsumā tika pieminētas iepriekš, šīm izceltajām metodēm ir būtiska nozīme, lai nodrošinātu sentimenta nolasīšanu precīzāk un efektīvāk. Atkarībā no konkrētā uzdevuma prasībām optimālu rezultātu sasniegšanai var izmantot šo metožu kombināciju. Svarīgi atcerēties, ka metodes izvēlei ir jāsaskan ar konkrēto sentimenta noteikšanas mērķi.

**1.2.3 Sentimenta nolasīšana**

Sentimenta nolasīšana ļauj noteikt emocionālo noskaņu, sajūtu, vai tā būtu pozitīva, negatīva vai neitrāla. Lai veiktu sentimenta noteikšanu ar mašīnmācīšanās palīdzību, tiek izmantoti dažādi algoritmi un tehniskie risinājum. Iepriekš apstrādātie dati, tiek padoti algoritmiem, kas analizē datus un veic sentimenta nolasīšanu. Šie algoritmi darbojas, apstrādājot tekstu, un analizējot tā sastāvdaļas, lai saprastu, kāda ir tā emocionālā noskaņa. Katrā algoritmā ir savas stiprās un vājās puses. [16.] Kopumā ir trīs veida pieejas kā noteikt sentimentu: **noteikumiem balstīta *(rule-based)*, automātiskā *(automatic)* un hibrīds *(hybrid)****.* [17]

**Noteikumiem balstīta pieeja *(rule-based)*:** Šis ir praktisks veids, kā analizēt tekstu bez nepieciešamības apmācīt vai izmantot mašīnmācīšanās modeļus. Šī pieejas rezultātā tiek izstrādāti noteikumi, pēc kuriem teksts tiek iezīmēts kā pozitīvs/negatīvs/neitrāls. Šie noteikumi ir pazīstami arī kā leksikoni. Tāpēc šo pieeju sauc par leksikonu balstītu pieeju. Plaši izmantoti noteikumu balstīti risinājumi ir *TextBlob* modelis, *VADER* modelis un algoritms, kā arī *SentiWordNet* modelis. [18]

Automātiskā pieeja *(automatic)*: Atšķirībā no noteikumu balstītās metodes automātiskā metode nenosaka manuāli izstrādātus noteikumus, bet gan izmanto mašīnmācīšanās tehnikas. Emocionālās analīzes uzdevums parasti tiek modelēts kā klasifikācijas problēma. Klasifikators saņem tekstu un nosaka kategoriju, piemēram, pozitīvu, negatīvu vai neitrālu. Piemēram, RoBERTa ir viens no populārākajiem modeļiem, kas izmanto automātisko pieeju. Kā arī polārākie algoritmi automātiskajai pieejai ir *Navie Bayes*, *Support Vector Machines (SVM)* un citi [19]

**Hibrīds *(hybrid)* pieeja:** Hibrīda pieeja apvieno abu pieeju veidu, gan noteikumu balstītu, gan automātisko tehniku, integrējot tos vienā pieejā. Viena no nozīmīgākajām priekšrocībām izmantojot šo pieeju, ir tas, ka tā spēj sniegt precīzākus rezultātus. [19]

Sentimenta precizitāti mēra ar dažādām metodēm kā piemēram: *F1-Score, Confusion Matrix, Cross-Validation* un citām metodēm. Pieejamie sentimenta noteikšanas veidi sniedz dažādas iespējas, kā veikt sentimenta nolasīšanu. Piemērotāko pieeju var izvēlēties pēc projekta nosacījumiem un datu kvalitātes.

**1.3 *VADER* un *RoBERTa* modeļu salīdzināšana**

VADER ir sentimenta lasīšanas modelis kā arī algoritms, taču algoritms nav publiski nekur minēts. Modelis ir daļa no *Python* bibliotēkas *nltk.sentiment* un tika speciāli radīts, lai strādātu ar sociālo tīklu tekstiem, bet, protams, tas spēj strādāt arī ar cita veida tekstiem. VADER iedala sentimentu polaritātes - negatīvs vai pozitīvs.

**1.4 Datu ievākšana no YouTube video komentāru sadaļas**

Datu iegūšana no kāda konkrēta *YouTube* video komentāru sadaļas ir svarīga, jo tā ļauj iegūt informāciju par sabiedrisko viedokli un reakcijām uz konkrētu video saturu. Tas var būt noderīgi, lai izprastu, kā cilvēki uztver, vai reaģē uz noteiktu tēmu vai notikumu, kā arī lai analizētu populārus viedokļus un tendences, kas izplatās sociālajā vidē. Pašlaik *YouTube* platformu lieto vairāk nekā 2 miljardi lietotāju katru mēnesi. Tāpēc *YouTube* video komentāros esošo informāciju var izmantot, lai veidotu mašīnmācīšanās modeļus, veiktu lietotāju sentimenta nolasīšanu, radītu lielas datu kopas un veiktu datu analīzes. Tas piedāvā plašas iespējas gan akadēmiskajos pētījumos, gan komerciālajā jomā. No juridiskās perspektīvas datu ievākšana, piemēram, komentāri un citi atribūti, ir pilnīgi likumīgs process. [20]

Vispopulārākais veids, kā iegūt datus no kāda konkrēta, video ir, izmantojot *YouTube data API v3*, ko piedāvā *Google*. *API* ir saīsinājums no angļu valodas *"Application Programming Interface"* jeb *"Pielietojuma programmēšanas saskarne"*. Tas ir programmēšanas protokols vai interfeiss, kas ļauj vienai programmai (vai daļai programmatūras) komunicēt ar citu programmu vai komponenti. API nosaka, kādā veidā programmatūras komponentes vai sistēmas var mijiedarboties un kādas darbības tās var veikt. [21] YouTube arī ir savs API, kuru var izmantot par velti un ar kura palīdzību ir iespējams iegūt dažādu informāciju no konkrēta *YouTube* video komentāru sadaļas. Kā, piemēram, ir iespējams iegūt komentārus, komentāru *“patīk”* balsu skaitu, komentāru atbildes, komentētāju lietotāj vārdus un citus datus par komentāru sadaļu. [22] Šo *API* var, izmanot ar Python un tās piedāvātajām bibliotēkām.

Taču ir arī citas metodes, piemēram, rasmošana *(web scraping)*. Rasmošana ir process, kurā programmētājs izmanto programmatūru, lai automātiski iegūtu informāciju no interneta lapām. Tas ietver HTML kodu analīzi un datu iegūšanu no tīmekļa vietnēm. Rasmošana var būt noderīga gadījumos, kad API nav pieejams vai neattiecas uz konkrēto informāciju, ko vēlaties iegūt. [23] Tomēr ir svarīgi atzīmēt, ka web scraping var pārkāpt YouTube lietošanas noteikumus specifiskos gadījumos. Ja tiek apskatīta konkrēta YouTube video komentāru sadaļa, tad nav optimāli izmantot rasmošanas metodi, jo datu iegūšana var, prasīs ilgu laiku. Tas prasa ilgu laiku tāpēc, ka šai programmai ir jāapstrādā katrs komentārs atsevišķi. [24]