UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET

Katedra za računarstvo

# Uticaj primene metoda augmentacije teksualnih podataka na detekciju govora mržnje

Seminarski rad

Predmet: Prikupljanje i predobrada podataka za mašinsko učenje

Mentor:

doc. dr Aleksandar Stanimirović

Student:

Katarina Stanojković, br. ind. 1773

Niš, 2024. godina

# Sadržaj

# 1. Uvod

U savremenom razvoju sistema mašinskog učenja, podaci predstavljaju osnovu za izgradnju modela koji mogu pouzdano obraditi i analizirati kompleksne zadatke. Kvalitet i kvantitet tih podataka direktno utiču na performanse modela, ali prikupljanje velike količine relevantnih podataka često predstavlja izazov, posebno kada je reč o tekstualnim podacima. Ovi podaci mogu biti ograničeni u obimu, raznovrsnosti i kvalitetu, što može dovesti do lošije generalizacije modela i problema sa overfitting-om.

Kako bi se prevazišli ovi izazovi, metode augmentacije tekstualnih podataka su se pokazale kao ključne tehnike za unapređenje raznovrsnosti i kvaliteta skupa podataka. Augmentacija tekstualnih podataka obuhvata generisanje novih primera na osnovu postojećih, pri čemu se čuvaju ključne semantičke informacije. Ove metode omogućavaju poboljšanje performansi modela i njihovu otpornost na varijacije u realnim podacima, dok istovremeno smanjuju rizik od prekomernog prilagođavanja (overfitting).

Cilj ovog rada je pružiti pregled različitih tehnika augmentacije tekstualnih podataka i analizirati njihov uticaj na rešavanje praktičnih problema u obradi prirodnog jezika (NLP). Poseban akcenat stavljen je na primenu ovih metoda u kontekstu detekcije govora mržnje, jedne od ključnih oblasti primene u savremenom NLP-u.

Rad uključuje i praktičan deo koji demonstrira implementaciju metoda augmentacije nad specifičnim skupom podataka, analizu njihovog uticaja na performanse različitih modela mašinskog učenja i uporednu evaluaciju rezultata. Ovaj pristup omogućava sticanje uvida u efikasnost različitih tehnika i pruža preporuke za njihovu primenu u stvarnim NLP zadacima.

# 1. Predobrada tekstualnih podataka

## 2.1. Čišćenje teksta

Čišćenje teksta je prvi korak u pripremi podataka za obradu, jer pomaže u uklanjanju nepotrebnih elemenata koji mogu ometati modele u tačnom prepoznavanju obrazaca u tekstu. Proces obično uključuje sledeće:

- **Uklanjanje znakova interpunkcije**: Interpunkcija poput tačaka, zareza, upitnika i uzvičnika često se uklanja jer obično ne doprinosi semantičkom značenju teksta u NLP zadacima.
- **Uklanjanje brojeva**: Brojevi se obično uklanjaju, osim u slučajevima kada su važni za analizu, poput analiza koje se bave numeričkim podacima.
- **Uklanjanje emotikona i specijalnih karaktera**: Emotikoni i drugi simboli kao što su "$", "&" ili "@", koji nisu deo semantičkog značenja, takođe se uklanjaju.
- **Uklanjanje URL-ova i email adresa**: Linkovi i email adrese često nisu korisni za analizu i mogu ometati tok obrade.
- **Uklanjanje HTML tagova**: U tekstovima preuzetim sa interneta ili iz HTML izvora, često se uklanjaju HTML tagovi, koji mogu dodavati šum.

Ove korake je potrebno prilagoditi zavisno od zadatka, jer se ponekad neki elementi mogu zadržati ako su relevantni za specifičnu analizu.

## 2.2. Normalizacija

Normalizacija teksta osigurava doslednost tako što transformiše tekst u standardizovani oblik. Glavni ciljevi normalizacije uključuju:

- **Pretvaranje velikih slova u mala slova**: Ovaj proces pomaže da se izbegne razlikovanje između reči poput "Pas" i "pas", čime se smanjuje broj varijacija koje model mora da nauči.
- **Uklanjanje dijakritika**: U jezicima poput srpskog, francuskog ili španskog, dijakritički znakovi (npr. č, ć, š, é) mogu biti uklonjeni kako bi se smanjile varijacije u rečima ("čovek" -> "covek").
- **Standardizacija teksta**: Uključuje zamenu različitih oblika reči njihovim doslednim varijantama (npr. "color" i "colour" postaju "color").

Normalizacija je važna za stvaranje doslednog skupa podataka koji će omogućiti bolju generalizaciju modela.

## 2.3. Tokenizacija

Tokenizacija je ključan proces u obradi prirodnog jezika, jer razbija tekst na manje jedinice (tokene) koje model može lakše obrađivati. Postoji nekoliko vrsta tokenizacije:

- **Tokenizacija na nivou reči**: Tekst se deli na pojedinačne reči. Ova vrsta tokenizacije se često koristi u zadacima poput klasifikacije teksta, analize sentimenta i pretraživanja informacija, gde je razumevanje pojedinačnih reči ključno. Na primer, za rečenicu „NLP is fun!“, tokenizacija na nivou reči bi dala tokene [„NLP“, „is“, „fun“, „!“].
- **Tokenizacija na nivou rečenica**: Tekst se deli na rečenice. Ova vrsta tokenizacije je korisna u zadacima poput sažimanja teksta i mašinskog prevođenja, gde je fokus na razumevanju i obradi celih rečenica. Na primer, za tekst „ NLP is fun. Let's learning it together.“, tokenizacija na nivou rečenica bi dala tokene [„NLP is fun.“, „Let's learning it together.“].
- **Tokenizacija na nivou karaktera**: Tekst se deli na pojedinačne karaktere. Tokenizacija na nivou karaktera se koristi u zadacima poput modelovanja jezika za jezike sa složenim pismima, generisanja teksta na nivou karaktera i rukovanja rečima koje nisu u vokabularu modela. Na primer, za reč „NLP“, tokenizacija na nivou karaktera bi dala tokene [„N“, „L“, „P“].
- **Tokenizacija na nivou subreči**: Tekst se deli na jedinice subreči, često korišćene u jezicima sa bogatom morfologijom. Ova vrsta tokenizacije je efikasna u neuronskom mašinskom prevođenju i rukovanju jezicima sa složenim oblicima reči, gde reči mogu imati više prefiksa i sufiksa. Na primer, za reč „unhapiness“, tokenizacija na nivou subreči bi dala tokene [„un“, „hapiness“].

Tokenizacija je ključna jer omogućava modelima da obrađuju tekst u delovima.

## 2.4. Lematizacija i stemovanje

Lematizacija i stemovanje su tehnike za redukciju reči na njihov osnovni oblik. Ove tehnike se koriste kako bi se smanjila varijacija u rečima koje predstavljaju iste koncepte.

- **Lematizacija**: Pretvara reč u njen osnovni oblik na temelju njenog značenja i konteksta. Na primer, "running", "ran" i "runs" bi se sve vratile na osnovni oblik "run". Lematizacija zahteva lingvističke informacije o reči, što je čini preciznijom od stemovanja.
- **Stemovanje**: Skraćuje reči uklanjanjem završetaka, ali bez uzimanja u obzir značenja reči. Na primer, "running" postaje "run", ali isto tako i reč "runner" postaje "run". Iako je brže i jednostavnije, stemovanje je manje precizno jer ne uzima u obzir gramatiku i kontekst.

Obe tehnike su korisne za smanjenje redundantnih oblika reči u tekstu, što olakšava modelima da uče efikasnije.

## 2.5. Uklanjanje stop-reči

Stop-reči su uobičajene reči kao što su "the", "is", "in", "at" koje se često pojavljuju u tekstu, ali ne nose mnogo informacija o značenju. Uklanjanje stop-reči omogućava modelima da se fokusiraju na reči koje nose veću informativnu vrednost.

- **Unapred definisane liste stop-reči**: Većina NLP biblioteka dolazi sa unapred definisanim listama stop-reči, ali se ove liste mogu prilagoditi potrebama zadatka.
- **Prilagođavanje stop-reči**: Ponekad je potrebno kreirati prilagođene liste stop-reči za specifične domene. Na primer, u medicinskim tekstovima reč "patient" može biti relevantna i ne bi trebala biti uklonjena.

Ovaj korak smanjuje šum u podacima i poboljšava performanse modela.

## 2.6. Spajanje kontrakcija

Spajanje kontrakcija podrazumeva proširivanje skraćenih oblika reči u njihov pun oblik. Kontrakcije poput "can't", "won't" ili "I'm" često zbunjuju modele za obradu teksta jer predstavljaju više reči spojenih u jednu.

- **Proširenje kontrakcija**: "can't" postaje "cannot", "I'm" postaje "I am", itd. Ovo olakšava precizniju tokenizaciju i bolji rad modela.
- **Poboljšana analiza teksta**: Kada se kontrakcije prošire, modeli za obradu teksta imaju tačniji prikaz onoga što tekst znači, što poboljšava sve daljnje korake u obradi.

Ovo je posebno važno u tekstovima koji sadrže puno kolokvijalnog jezika ili neformalnih izraza.

## 2.7. Uklanjanje duplikata

Uklanjanje duplikata je proces kojim se iz skupa podataka uklanjaju tekstovi koji se pojavljuju više puta. Ovaj korak je važan kako bi se izbegla redundantnost koja može dovesti do prekomernog prilagođavanja modela (overfitting).

- **Smanjenje redundancije**: Duplikati često stvaraju problem kada model uči iz skupa podataka, jer isti podaci više puta utiču na rezultate.
- **Povećana efikasnost**: Kada se duplikati uklone, dataset postaje manji i brži za obradu, što poboljšava ukupne performanse modela.

## 2. Pregled tehnika augmentacije tekstualnih podataka

Augmentacija teksta obuhvata širok spektar metoda, koje se mogu primeniti na različitim nivoima tekstualnih podataka, od reči i fraza, do celih dokumenata. Kroz ovaj rad, fokusiraćemo se na pregled različitih tehnika augmentacije tekstualnih podataka, sa naglaskom na njihov uticaj na detekciju govora mržnje.

U ovom kontekstu, tehnike augmentacije se mogu podeliti u dve glavne kategorije: one koje se primenjuju na nivou karakteristika (Feature Space) i one koje direktno manipulišu tekstualnim instancama (Data Space). Pregled ovih metoda obuhvata jednostavne pristupe, kao što su sinonimna zamena i back-translation, kao i napredne metode koje koriste jezičke modele i generativne algoritme. Ovaj pregled pruža uvid u ključne prednosti i ograničenja svake od ovih tehnika, istovremeno naglašavajući specifičnosti njihove primene u realnim zadacima obrade teksta.

Kroz poređenje ovih tehnika, rad nudi smernice o tome koja metoda augmentacije je najpogodnija za određene situacije, osvetljavajući ključne aspekte koji utiču na performanse modela u različitim kontekstima.

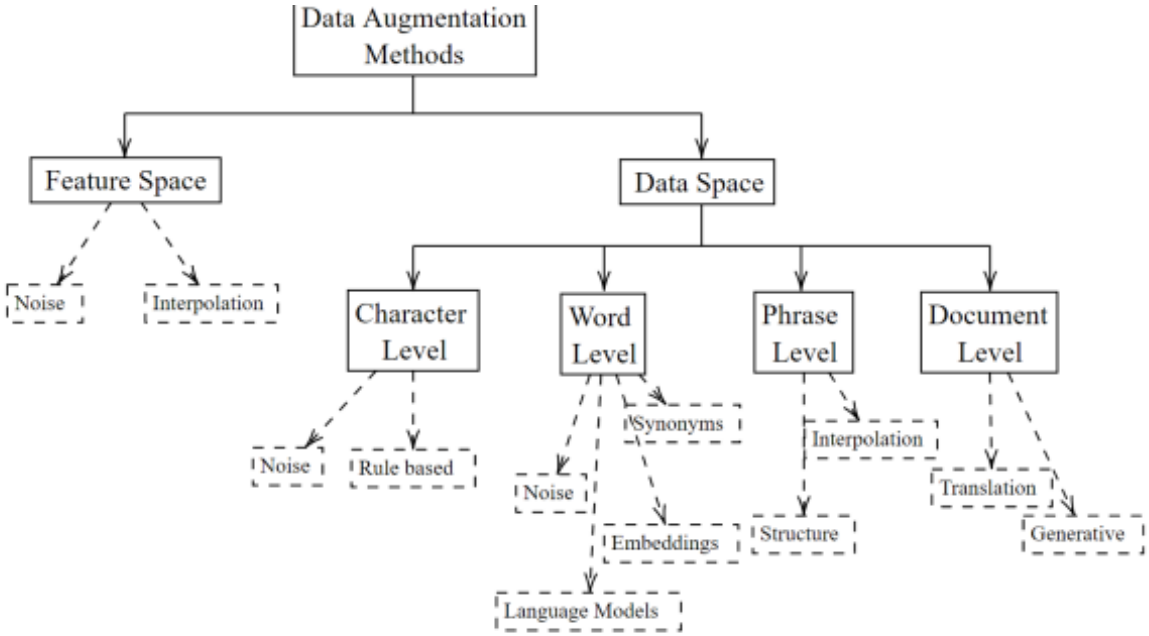



Figure 1: Taxonomy and grouping for different data augmentation methods.

## 3.1. Data Space

### 3.1.1. Na nivou karaktera

Augmentacija tekstualnih podataka na nivou karaktera podrazumeva različite transformacije koje se primenjuju na najmanjoj jedinici teksta – karakterima. Metode augmentacije na nivou karaktera često koriste pravila ili uvode šum kako bi simulirale greške koje se javljaju prilikom kucanja, skeniranja ili prepoznavanja teksta.

#### *3.1.1.1. Simulacija pravopisnih grešaka*

Ova tehnika koristi unapred definisana pravila kako bi stvorila greške koje podsećaju na uobičajene pravopisne pogreške. Ova metoda omogućava modelu da nauči da prepoznaje varijante reči sa pravopisnim greškama, čime se povećava njegova sposobnost da pravilno interpretira tekst sa greškama.

**"The fox jumps over the lazy dog" → "The fox jumps over the lazi dog"**

#### *3.1.1.2. Simulacija grešaka u kucanju (Keyboard Augmenter)*

Ova metoda simulira greške u kucanju na osnovu blizine tastera na tastaturi, što imitira uobičajene tipografske greške koje se dešavaju pri brzom kucanju. Ova augmentacija pomaže modelu da postane otporniji na greške koje korisnici prave prilikom unosa teksta.

**"The fox jumps over the lazy dog" → "The fox jumps over the lazu dog"**

#### *3.1.1.3. Optičko prepoznavanje karaktera (OCR simulacija)*

Ova tehnika imitira greške koje nastaju prilikom korišćenja sistema za optičko prepoznavanje karaktera (OCR), gde slični karakteri često bivaju pogrešno interpretirani. Ova metoda je korisna za treniranje modela da prepoznaje tekst iz različitih skeniranih dokumenata sa greškama u prepoznavanju.

**"The fox jumps over the lazy dog" → "The fox jumps over the 1azy dog"**

#### *3.1.1.4. Random augmentacija karaktera*

Ova tehnika uvodi nasumične varijacije u tekstualne podatke. Postoje 4 osnovne operacije:

- Insert – Ova operacija dodaje karaktere na nasumičnim mestima u tekstu
  **"Boat" → "Boaat"**
- Substitute – Ova operacija vrši slučajnu zamenu postojećeg karaktera drugim
  **"Boat" → "Beat"**
- Swap – Ova operacija slučajno menja mesta dva karaktera
  **"Boat" → "Baot"**
- Delete – Ova operacija slučajno briše postojeći karakter
  **"Boat" → "Bat"**

### 3.1.2. Na nivou reči

Augmentacija na nivou reči obuhvata tehnike koje modifikuju pojedinačne reči u rečenici kako bi generisale varijacije teksta, a istovremeno očuvale osnovni smisao. Ove metode često koriste zamene sinonima, indukciju šuma ili naprednije metode kao što su ugneždena predstavljanja reči (embedding). Cilj je povećanje raznovrsnosti podataka za treniranje modela, što doprinosi boljoj generalizaciji i robusnosti modela u realnim uslovima.

#### *3.1.2.1. Zamena sinonima*

Sinonimna zamena koristi tezaurus kao što je WordNet za pronalaženje semantički sličnih reči. Ova metoda koristi verovatnoću zamene na osnovu distribucije, gde se retki sinonimi preferiraju, što doprinosi boljem učenju, posebno kod zadataka s malim resursima.

**"Happy" → "Joyful"**

#### *3.1.2.2. Zamena antonima:*

Zamena antonima menja reči sa njihovim suprotnim značenjima. Ova metoda uvodi kontrast u tekst i može biti korisna za generisanje raznovrsnijih konteksta ili za testiranje sposobnosti modela da prepozna promene u sentimentu.

**"Happy" → "Sad"**

#### *3.1.2.3. Umetanje reči (Random Insertion):*

Nasumično umetanje reči dodaje dodatne ili semantički slične reči u rečenicu, bez promene njenog osnovnog značenja. Ova tehnika dodaje detalje tekstu, povećavajući njegovu bogatost i raznovrsnost.

**"He enjoys reading books" → "He really enjoys reading interesting books"**

#### *3.1.2.4. Nasumična zamena (Random Swap):*

Nasumična zamena menja pozicije reči unutar rečenice, zadržavajući njihov semantički smisao. Ova metoda unosi varijacije u strukturu rečenice, što pomaže modelu da postane fleksibilniji u prepoznavanju različitih redosleda reči.

**"The quick brown fox jumps over the lazy dog"**
**→ "The brown quick fox jumps over the lazy dog"**

#### *3.1.2.5. Nasumično brisanje (Random Deletion):*

Nasumično brisanje uklanja pojedine reči iz rečenice, ali se zadržava njen osnovni smisao. Ova tehnika može skratiti rečenice i pomoći modelu da nauči da radi sa nepotpunim podacima.

**"Artificial intelligence is transforming the world"**
**→ "Artificial intelligence is transforming world"**

#### *3.1.2.6. Podela reči (Split Augmentation):*

Podela reči nasumično deli reči na manje delove, menjajući strukturu reči i stvarajući nove tekstualne varijacije. Ova metoda pomaže u povećanju varijabilnosti u trening podacima.

**"Darkness" → "Dark ness"**

#### *3.1.2.7. Rezervisane reči (Reserved Word Augmentation):*

Ova tehnika omogućava definisanje reči koje se ne smeju menjati tokom augmentacije. Na primer, ukoliko je važno da reč "lion" ostane nepromenjena, može se koristiti lista rezervisanih reči kako bi se izbegla njena zamena tokom procesa augmentacije.

**"The lion roared loudly" → "The lion roared extremely loudly"**

(ako je "lion" rezervisana reč, ostaje nepromenjena).

#### *3.1.2.8. TF-IDF Augmentacija:*

Ova metoda koristi termine sa niskim TF-IDF skorom, koje se mogu zameniti drugim rečima sa sličnim skorom. TF-IDF (Term Frequency-Inverse Document Frequency) izračunava važnost reči u tekstu, omogućavajući zamenu manje značajnih reči bez promene osnovnog značenja.

**"The cat sat on the mat" → "The cat sat on the rug"**

### 3.1.3. Na nivou fraza i rečenica

Augmentacija na nivou fraza i rečenica fokusira se na modifikaciju celokupnih fraza ili rečenica kako bi se stvorile nove varijante teksta uz očuvanje semantičkog značenja. Ove tehnike omogućavaju generisanje varijacija u tekstu koje pomažu u diversifikaciji skupa podataka i unapređuju sposobnost modela da generalizuje na realnim primerima.

#### *3.1.3.1. Kropljenje i rotacija (Cropping and Rotation)*

Kropljenje i rotacija su tehnike inspirisane obradom slike, primenjene na strukturu rečenice. Kropljenje podrazumeva skraćivanje rečenice fokusiranjem na ključne komponente, kao što su subjekti ili objekti.

**"The cat sat on the mat and looked at the dog" → "The cat sat on the mat"**

Rotacija podrazumeva premeštanje delova rečenice oko ključnog korena, stvarajući različite varijante.

**"The cat sat on the mat" → "On the mat sat the cat"**

Ove tehnike su korisne za zadatke kao što je označavanje delova govora.

#### *3.1.3.2. Zamena fraza pomoću zavisnih stabala*

Ova metoda koristi zavisna stabla kako bi se identifikovale fraze koje mogu biti zamenjene sličnim frazama iz drugih rečenica. Zamena fraza omogućava generisanje novih sintetičkih tekstova koji zadržavaju gramatiku i strukturu originalne rečenice, ali uvode semantičku raznolikost.

**"The quick brown fox jumps over the lazy dog"**
**→ "The quick brown fox jumps over the sleepy cat"**

#### *3.1.3.3. Nasumična augmentacija rečenica (Random Sentence Augmentation)*

Nasumična augmentacija na nivou rečenica uključuje različite nasumične operacije, kao što su brisanje, premeštanje ili zamena rečenica unutar većeg teksta. Ove varijacije stvaraju nove verzije teksta koje zadržavaju osnovni smisao, ali menjaju redosled ili strukturu rečenica, što doprinosi diversifikaciji trening skupa podataka.

**"It was a dark and stormy night. I was alone at home when I saw a lion's face followed by a scary thunderous roar at the windows"**
**→ " I was alone at home when I saw a lion's face followed by a scary thunderous roar at the windows. It was a dark and stormy night"**

### 3.1.4. Na nivou dokumenta

Augmentacija na nivou dokumenta obuhvata tehnike koje se primenjuju na celu jedinicu teksta, odnosno na kompletne dokumente. Ove metode omogućavaju generisanje novih dokumenata na osnovu originalnih, čime se povećava raznovrsnost i količina podataka za treniranje modela. Tehnike na nivou dokumenta često uključuju prevođenje, korišćenje generativnih modela i druge napredne metode koje stvaraju sintetičke dokumente sa očuvanim semantičkim značenjem.

#### *3.1.4.1. Back-Translation (BT)*

Back-Translation je metoda koja koristi prevod sa jednog jezika na drugi, a zatim vraćanje teksta na originalni jezik kako bi se dobila nova sintetička verzija. Ova tehnika generiše parafrazirani tekst koji zadržava osnovno značenje originala, ali uvodi leksičke i sintaksne varijacije.

**"The cat sat on the mat"**
**→ "Le chat s'est assis sur le tapis"**
**→ "The cat rested on the rug"**

Back-Translation je naročito korisna u zadacima mašinskog prevođenja, ali je pokazala i uspeh u drugim NLP zadacima poput analize sentimenta i odgovaranja na pitanja, gde doprinosi povećanju količine i raznolikosti trening podataka.

#### *3.1.4.2. Iterative Back-Translation (IterativeBT):*

Iterative Back-Translation predstavlja unapređenu varijantu osnovne Back-Translation metode, gde se proces prevođenja i vraćanja teksta ponavlja više puta. Svaka iteracija koristi unapređene prevode generisane prethodnim modelom, čime se model kontinuirano poboljšava u generisanju sintetičkog teksta.

**"The cat sat on the mat"**
**→ "El gato se sentó en la alfombra"**
**→ "The cat sat on the doormat"**

**→ "Die Katze saß auf der Matte"**
**→ "The cat was sitting on the mat"**

Ovaj pristup je posebno koristan u okruženjima sa malim resursima, jer omogućava generisanje većih količina podataka za treniranje modela bez potrebe za dodatnim ručnim unosom.

#### *3.1.4.3. Noised Back-Translation (NoisedBT):*

Noised Back-Translation kombinuje osnovnu Back-Translation tehniku sa dodavanjem šuma u tekst tokom procesa prevođenja. Šum može uključivati nasumično brisanje, zamenu ili premeštanje reči, čime se dodatno povećava raznovrsnost sintetičkog teksta.

**"The cat sat on the mat"**
**→ "The cat on sat the mat"**
**→ "Il gatto era seduto sul tappetino"**

**→ "The cat was sitting on the carpet"**

Dodavanje šuma čini model otpornijim na prirodne varijacije u tekstu, omogućavajući mu da bolje prepoznaje i interpretira različite strukture rečenica.

#### *3.1.4.4. Tagged Back-Translation (TaggedBT):*

Tagged Back-Translation je varijanta Back-Translation tehnike koja koristi specijalne tagove za označavanje sintetički generisanog teksta. Umesto dodavanja šuma, ovaj pristup označava generisane podatke kako bi model naučio da razlikuje originalni i sintetički tekst.

**"The cat sat on the mat"→ "[SYN] The cat rested on the rug"**

Ova tehnika omogućava modelu da ispravno koristi sintetičke podatke, čime se poboljšava tačnost i robusnost modela bez narušavanja semantičkog značenja originalnog teksta.

#### *3.1.4.5. Back Transliteration*

**Back Transliteration** je metoda augmentacije podataka koja se koristi za generisanje rečenica ili fraza koje zvuče fonetski slično izvornom jeziku, ali su napisane u drugom pismu. Ova tehnika je posebno korisna za generisanje trening podataka za klasifikacione zadatke koji uključuju lokalizovane ili bi-jezične fraze, gde je ciljni jezik jezik sa malim resursima, odnosno ima manje dostupnih izvora podataka.

**"Machine learning is a subset of AI"→ "Машине лернинг ис а сабсет оф АИ"**

## 3.2. Feature Space

Augmentacija podataka u **feature space** (prostoru karakteristika) znači da se ne rade promene direktno na tekstu, već na njegovim numeričkim reprezentacijama, tj. na vektorskim prikazima rečenica ili reči (tzv. embeddings). Ove metode omogućavaju da se kreiraju nove varijacije podataka bez promene originalnog teksta, čime se model trenira da bude robusniji i otporniji na različite varijacije.

### 3.2.1. Indukcija šuma

Indukcija šuma u prostoru karakteristika podrazumeva dodavanje malih, nasumičnih promena na vektorskim prikazima teksta. Ove promene pomažu modelu da uči iz raznih varijacija i da postane bolji u prepoznavanju sličnih podataka.

Umesto da se menja stvarni tekst, dodaju se male nasumične promene u vektorskim prikazima rečenica. Na primer, ako se rečenica "The cat is on the mat" predstavi vektorima, šum može dodati male promene u te brojeve, ali će osnovno značenje ostati isto. Model se trenira na takvim promenjenim podacima kako bi postao otporniji na greške ili manipulacije.

### 3.2.2. Interpolacione metode

Interpolacione metode prave nove podatke kombinovanjem dve ili više rečenica, ali ne direktno, već koristeći njihove numeričke prikaze (embeddings). Ovo pomaže modelima da bolje generalizuju i da budu otporniji na prekomerno prilagođavanje (overfitting).

#### *3.2.2.1. SMOTE Interpolacija*

**SMOTE** je metoda koja pomaže da se balansiraju podaci u zadacima klasifikacije. Umesto da jednostavno kopira postojeće podatke, SMOTE kreira nove instance tako što kombinuje slične instance iz iste klase.

**Primer**: Ako imamo dve slične rečenice iz iste klase, SMOTE će stvoriti novu rečenicu kombinujući njihove vektorske prikaze. To pomaže u balansiranju skupa podataka.

#### *3.2.2.2. Mixup Interpolacija*

**Mixup** je metoda koja kombinuje dve različite rečenice i njihove klase kako bi stvorila novu rečenicu koja predstavlja neku vrstu "mešavine" obe. Ova metoda pomaže modelu da uči iz različitih klasa i da bolje generalizuje.

**Primer**: Ako imamo rečenicu iz klase 0 ("The cat is on the mat") i rečenicu iz klase 1 ("The dog is barking"), mixup tehnika će stvoriti novu rečenicu koja je kombinacija obe, a i klasa će biti mešavina oba originalna labela.

Mixup se često koristi u dubokim modelima (kao što su BERT ili RoBERTa), gde se ove kombinacije primenjuju na različitim slojevima mreže, čime se poboljšava razumevanje složenih obrazaca u tekstu.

# 4. Napredne metode augmentacije podataka

## 4.1. Generativni modeli

### *4.1.1. GPT-based modeli*

Generative Pre-trained Transformer (GPT) modeli, predstavljaju značajan napredak u oblasti obrade prirodnog jezika (NLP). GPT-2 i GPT-3, su autoregresivni modeli jezika koji koriste duboko učenje za generisanje teksta koji je sličan ljudskom. Ovi modeli su prethodno trenirani na ogromnim skupovima podataka, što im omogućava da razumeju i proizvode koherentan i kontekstualno relevantan jezik u različitim temama i stilovima.

GPT-2, sa svojih 1,5 milijardi parametara, bio je revolucionaran model zahvaljujući svojoj sposobnosti da generiše tekst na osnovu datog prompta. Njegova sposobnost razumevanja konteksta i proizvodnje relevantnih nastavaka čini ga neprocenjivim alatom za augmentaciju teksta.

Nadograđujući osnove GPT-2, GPT-3 značajno unapređuje sposobnosti modela sa svojih 175 milijardi parametara, čineći ga jednim od najvećih i najmoćnijih modela jezika do danas. GPT-3-ova poboljšana kapacitet omogućava preciznije i kontekstualno svesnije generisanje teksta, olakšavajući još veću raznolikost i kvalitet u augmentisanim skupovima podataka.

Sposobnost GPT-3 da generiše veliki broj raznovrsnih i kontekstualno prikladnih varijacija teksta čini ga esencijalnim alatom za unapređenje skupova podataka. Ova povećana raznolikost ne samo da smanjuje probleme vezane za prekomerno prilagođavanje (overfitting), već oprema modele mašinskog učenja bogatijim razumevanjem jezičkih nijansi, što na kraju vodi ka preciznijim i pouzdanijim predviđanjima.

GPT-based augmentacija nudi nekoliko ključnih prednosti, uključujući skalabilnost, jer GPT modeli mogu brzo generisati velike količine teksta, ovi modeli takođe uvode raznolikost u formulacijama i strukturama, održavanje kontekstualne relevantnosti osigurava da generisani podaci ostaju semantički integrisani i značajni za originalni tekst. Međutim, korišćenje GPT-based augmentacije nosi sa sobom i određene izazove. Prvo, modeli su resursno intenzivni, jer treniranje i implementacija velikih GPT modela zahteva značajne računarske resurse; kontrola kvaliteta je neophodna, jer generisani tekst može ponekad sadržati netačan ili irelevantan sadržaj, što zahteva pažljiv pregled i filtriranje; etička pitanja su ključna, jer korišćenje generativnih modela mora biti pažljivo upravljano kako bi se sprečilo kreiranje pristrasnog ili neetičkog sadržaja.

### *4.1.2. GAN*

Generative Adversarial Networks (GANs) predstavljaju moćnu arhitekturu u veštačkoj inteligenciji. GAN-ovi se sastoje od dva konkurentna modela: generatora, koji kreira

sintetičke podatke slične stvarnim, i diskriminatora, koji pokušava da razlikuje stvarne podatke od generisanih. Ovaj sukob omogućava da se oba modela iterativno unapređuju, postavljajući visok nivo realističnosti generisanih podataka.

U kontekstu augmentacije tekstualnih podataka, GAN-ovi omogućavaju generisanje sintetičkih primera koji zadržavaju semantičko značenje originalnih podataka, dok uvode leksičke i sintaktičke varijacije.

Međutim, korišćenje GAN-ova za tekstualnu augmentaciju nosi izazove poput visokih računarskih zahteva, nestabilnosti treniranja i problema sa mode collapse, što može smanjiti raznolikost generisanih podataka. Takođe, generisani tekstovi mogu sadržavati artefakte ili neodgovarajući sadržaj, što zahteva dodatnu filtraciju i validaciju.

## 4.2. Contextual Embeddings

### *4.2.1. BERT-based augmentacija*

BERT (Bidirectional Encoder Representations from Transformers) je model koji koristi transformers arhitekturu za generisanje kontekstualizovanih rečničkih prikaza. Za razliku od unidirekcionalnih modela koji analiziraju tekst samo s leva na desno ili s desna na levo, BERT koristi bidirekcionalni pristup, što mu omogućava da uzme u obzir oba konteksta reči u rečenici. Ovo omogućava preciznije i kontekstualno prilagođene zamene i umetanja reči, čime se generiše kvalitetniji augmentovani tekst.

BERT-based augmentacija predstavlja naprednu metodu za proširenje skupa podataka korišćenjem kontekstualnih rečničkih prikaza koje generiše BERT model. Tradicionalni word embeddings dodeljuju statičan vektor svakoj reči, nezavisno od njenog konteksta, što može biti ograničavajuće u situacijama gde ista reč ima različita značenja u različitim kontekstima. Na primer, reč "Fox" može označavati životinju ili televizijsku kompaniju, a statičan vektor ne može adekvatno odražavati ovu raznolikost. Da bi se prevazišao ovaj problem, uvedena je kontekstualizovana word embeddings metoda, koja koristi okolne reči za generisanje specifičnih vektora u zavisnosti od konteksta u kojem se reč pojavljuje.

Jedan od alata za implementaciju BERT-based augmentacije je BertAug, koji je dizajniran da omogući umetanje i zamenu reči koristeći BERT jezički model. Za razliku od prethodnih metoda, umetanje reči se ne vrši nasumičnim izborom reči iz vokabulara, već se predviđa odgovarajuća reč na osnovu konteksta pomoću BERT modela. Slično tome, zamena reči koristi okolne reči kao karakteristike za predviđanje ciljne reči, čime se osigurava da zamena bude semantički i sintaktički pravilna.

### *4.2.2. RoBERTa, XLNet i drugi modeli*

**RoBERTa** je razvijen kao unapređenje BERT modela, optimizujući proces treniranja kako bi postigao bolje performanse na raznim NLP zadacima. Ključne razlike uključuju korišćenje veće količine podataka za treniranje, duže treniranje i uklanjanje Next Sentence Prediction (NSP) zadatka, fokusirajući se isključivo na Masked Language Model (MLM). Korišćenjem RoBERTa modela u augmentaciji teksta, moguće je generisati semantički bogatije i raznovrsnije varijacije rečenica, čime se povećava kvalitet i raznolikost skupa podataka.

**XLNet** predstavlja značajan napredak u razvoju transformera. Ključne karakteristike uključuju permutacijsko treniranje, gde se reči u rečenici permutuju pre treniranja, omogućavajući modelu da uči zavisnosti između reči u različitim redosledima. Ovo omogućava bolje hvatanje dugoročnih zavisnosti i izbegavanje problema maskiranog jezika karakterističnih za BERT. Korišćenjem XLNet-a za augmentaciju teksta, moguće je generisati tekstualne varijacije koje su semantički i sintaktički bogate, čime se poboljšava raznolikost skupa podataka.

Pored RoBERTa i XLNet-a, postoje i drugi napredni modeli koji koriste kontekstualizovana word embeddings za augmentaciju teksta:

- **DistilBERT i DistilRoBERTa**: Ovi modeli su kompaktne verzije originalnih BERT-a i RoBERTa modela, razvijene korišćenjem tehnike distilacije. Distilacija je proces u kojem se manji model (student) trenira da replicira ponašanje većeg modela (učitelj), čime se smanjuje broj parametara i ubrzava inferencija, a da se pritom zadrže slične performanse. DistilRoBERTa, na primer, omogućava efikasniju primenu u okruženjima sa ograničenim resursima bez značajnog gubitka tačnosti.
- **ALBERT (A Lite BERT)**: Model koji smanjuje broj parametara kroz faktorizaciju embedding matrica i deljenje težina između slojeva, omogućavajući brže treniranje i smanjenje memorijskih zahteva.
- **ERNIE (Enhanced Representation through kNowledge Integration)**: Model koji integriše spoljne izvore znanja tokom treniranja, poboljšavajući razumevanje semantičkih veza i odnosa između reči.

Ovi modeli pružaju različite prednosti u zavisnosti od specifičnih zahteva aplikacije, omogućavajući fleksibilnu i efikasnu augmentaciju teksta za unapređenje performansi NLP modela.

## 4.3. Parafraziranje teksta korišćenjem T5

**Text-to-Text Transfer Transformer (T5)** predstavlja naprednu tehniku za augmentaciju teksta koristeći veliki transformator model koji je treniran na C4 datasetu. Razvijen od strane

Google-a, T5 model je otvoreno dostupnog tipa i sposoban je za obavljanje različitih NLP zadataka kao što su prevođenje, sumarizacija, odgovaranje na pitanja i klasifikacija.

T5 model reframira svaki NLP zadatak u "text-to-text" format, što znači da svaki zadatak može biti izražen kao transformacija jednog teksta u drugi. Ova univerzalna arhitektura omogućava T5 modelu da bude veoma fleksibilan i prilagodljiv različitim zadacima bez potrebe za značajnim promenama u strukturi modela.

## 4.4. CLARE Augmenter

CLARE Augmenter je napredna tehnika za kreiranje parafraziranih tekstova koristeći kontekstualne izmene. CLARE, što je skraćenica za **ContextuaLized AdversaRial Example generation** model, koristi pre-trenirane modele maskiranog jezika poput RoBERTa kako bi generisao prirodne, tečne i gramatički ispravne tekstove. Ovi tekstovi služe kao dodatni podaci za različite NLP zadatke.

CLARE Augmenter koristi mask-then-infill pristup kroz četiri ključna koraka:

1. **Identifikacija ranjivih mesta**: Analizom delova govora (POS tagova) pronalaze se reči ili fraze koje je moguće izmeniti, posebno imenice i fraze imenica koje su važne za klasifikaciju teksta.
2. **Primena modifikacija** : Na identifikovana mesta se primenjuje neka od tri vrste izmena:
    - **Replace (Zamena)**: Zamenjuje postojeću reč odgovarajućom alternativom koja se uklapa u kontekst.
    - **Insert (Umetanje)**: Dodaje novu reč na način koji je smislen i prirodan u rečenici.
    - **Merge (Spajanje)**: Spaja dve reči ili fraze u jednu, menjajući strukturu rečenice.
3. **Odabir najboljih kandidata**: Generisane alternative se rangiraju prema verovatnoći koju dodeljuje model maskiranog jezika. Izabira se ona reč koja najviše smanjuje verovatnoću originalne klase kod ciljanog modela.
4. **Generisanje adversarijalnih primera**: Primenom odabranih izmena stvara se novi tekst koji je sličan originalu, ali dovoljno drugačiji da može izazvati grešku u ciljanom modelu.

# 5. Praktični deo rada

## 5.1. Opis dataset-a

Ovaj rad se fokusira na analizu i detekciju govora mržnje i uvredljivog jezika na Twitter-u korišćenjem posebnog skupa podataka nazvanog *hate_speech_offensive*. Dataset je pažljivo kreirana kolekcija anotiranih tvitova na engleskom jeziku, namenjena treniranju mašinskih modela za automatsko prepoznavanje govora mržnje i uvredljivog sadržaja. Skup podataka je dostupan u formatu CSV datoteke pod nazivom *train.csv* i nije podeljen na više delova—dostupan je samo trening set.

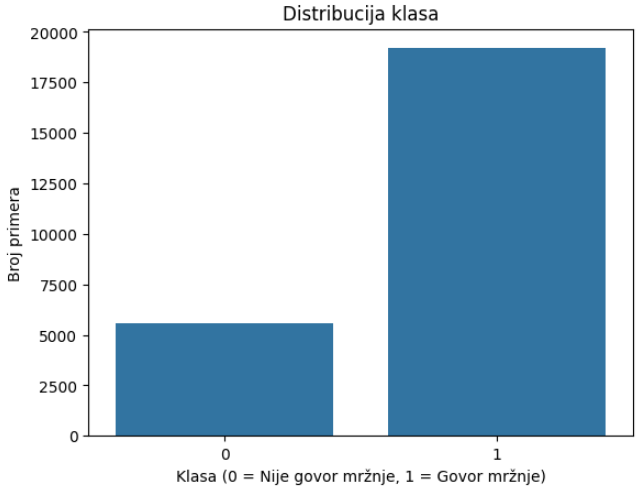
**Pregled skupa podataka**

Dataset sadrži nekoliko ključnih kolona koje pružaju detaljne informacije o klasifikaciji svakog tvita:

- **count**: Ukupan broj anotacija za svaki tvit.
- **hate_speech_count**: Broj anotacija koje klasifikuju tvit kao govor mržnje.
- **offensive_language_count**: Broj anotacija koje označavaju tvit kao uvredljiv jezik.
- **neither_count**: Broj anotacija koje identifikuju tvit kao ni govor mržnje ni uvredljiv jezik.
- **class**: Konačna vrednost klase, 0 – ne pripada ni govoru mržnje ni uvredljivom govoru, 1 – pripada govoru mržnje, 2 – pripada uvredljivom govoru.

Podaci su prikupljeni putem javnog Twitter API-ja korišćenjem specifičnih ključnih reči povezanih sa govorom mržnje i uvredljivim jezikom. Nakon prikupljanja, tvitove je ručno anotiralo više anotatora koji su ih klasifikovali u odgovarajuće kategorije.

Radi jednostavnosti, u ovom projektu je fokus na detekciji govora mržnje, stoga su klase 0 i 2 objedinjene u jednu – nije govor mržnje.



Možemo primetiti da je dataset u početnom trenutku nebalansiran, pa ćemo primenjivati metode za augmentaciju podataka na klasi 0 (nije govor mržnje), kako bismo povećali broj instance te klase i samim tim dobili balansiran dataset.

## 5.2. Preprocesiranje podataka

U cilju efikasnog treniranja modela za detekciju govora mržnje i uvredljivog jezika, neophodno je sprovesti detaljno preprocesiranje tekstualnih podataka. Predobrada teksta omogućava uklanjanje nepotrebnih informacija i normalizaciju podataka, što poboljšava performanse mašinskih algoritama.

**Koraci preprocesiranja:**

1. **Uklanjanje URL-ova i email adresa:** URL-ovi i email adrese ne doprinose semantičkom sadržaju tvita u kontekstu detekcije govora mržnje i mogu uneti šum u model. Korišćenjem regularnih izraza uklanjaju se sve instance URL-ova (počinju sa `http`, `www`) i email adresa.
2. **Uklanjanje HTML tagova:** HTML tagovi su tehnički elementi koji nisu relevantni za analizu teksta i mogu ometati procesiranje. Biblioteka **BeautifulSoup** koristi se za parsiranje i uklanjanje bilo kakvih HTML tagova koji mogu biti prisutni u tekstu.
3. **Uklanjanje emotikona i specijalnih karaktera:** Iako emotikoni mogu nositi emocionalni ton, često otežavaju procesiranje zbog kodiranja i mogu biti izvor šuma. Definiše se funkcija `remove_emojis` koja koristi regularne izraze za identifikaciju i uklanjanje emotikona i specijalnih simbola.
4. **Uklanjanje specijalnih karaktera i interpunkcije:** Cilj je zadržati samo korisne tekstualne informacije i eliminisati potencijalno nerelevantne simbole. Uklanjaju se svi karakteri koji nisu slova (uključujući slova sa dijakritičkim znacima) i razmaci.
5. **Pretvaranje u mala slova:** Uniformisanje teksta olakšava procesiranje i osigurava da se reči poput "Govor" i "govor" tretiraju isto. Cela tekstualna kolona se konvertuje u mala slova.
6. **Uklanjanje dijakritika:** Normalizacija teksta za potrebe modela koji možda ne prepoznaju dijakritike, čime se smanjuje kompleksnost vokabulara. Korišćenjem biblioteke **unicodedata**, uklanjaju se dijakritički znakovi iz slova.
7. **Uklanjanje višestrukih razmaka:** Čisti tekst od nepotrebnih praznina koje mogu uticati na tokenizaciju. Višestruki razmaci se zamenjuju jednim razmakom, a vodeći i prateći razmaci se uklanjaju.
8. **Tokenizacija:** Tokenizacija je neophodan korak za mnoge NLP procese, omogućava individualnu obradu svake reči. Tekst se deli na individualne reči (tokene) korišćenjem biblioteke **nltk**.

9. **Uklanjanje stop-reči:** Stop-reči mogu zamagliti značajne obrasce u podacima; njihovo uklanjanje fokusira analizu na informativne reči. Iz tokena se uklanjaju uobičajene reči koje ne doprinose značenju.
10. **Lematizacija:** Lematizacija pomaže u smanjenju varijacija reči, što omogućava modelu da prepozna iste reči u različitim oblicima. Reči se svode na njihov osnovni ili korenski oblik korišćenjem **WordNetLemmatizer**.

## 5.3. Primena modela nad osnovnim dataset-om

Korišćene metode uključuju klasične algoritme mašinskog učenja kao što su logistička regresija, mašina sa podrškom vektora (SVM) i Naive Bayes, kao i duboke neuronske mreže poput LSTM i CNN.

1. **Logistička regresija**

Logistička regresija je statistički model koji se koristi za binarnu klasifikaciju. Ona procenjuje verovatnoću da određeni ulaz pripada jednoj od dve moguće kategorije. U ovom slučaju, koristi se za predviđanje da li tvit sadrži govor mržnje (klasa 1) ili ne (klasa 0). Ovaj model je jednostavan za implementaciju i interpretaciju, što ga čini čestim izborom u NLP zadacima.

2. **SVM**

SVM je nadgledani algoritam za mašinsko učenje koji se koristi za klasifikaciju i regresiju. Cilj SVM-a je pronaći optimalnu hiper-ravan koja najbolje razdvaja klase u visokodimenzionalnom prostoru. U tekstualnoj klasifikaciji, SVM je efikasan zbog svoje sposobnosti da radi sa velikim brojem karakteristika koje proizilaze iz vektorizacije teksta.

3. **Naive Bayes klasifikator**

Naive Bayes je probabilistički klasifikator zasnovan na Bajesovoj teoremi, uz pretpostavku da su sve karakteristike međusobno nezavisne. Ova pretpostavka pojednostavljuje model i čini ga efikasnim za velike skupove podataka. Često se koristi u NLP zbog svoje brzine i efikasnosti, posebno kod problema klasifikacije teksta.

4. **Long Short-Term Memory (LSTM) neuronske mreže**

LSTM je tip rekurentne neuronske mreže (RNN) koja je dizajnirana da prevaziđe problem kratkoročne memorije u RNN-ovima. LSTM ima unutrašnje mehanizme, poput ćelijskih stanja i vrata, koji omogućavaju čuvanje informacija kroz duge sekvence. Ovo je posebno korisno u obradi teksta, gde kontekst i sekvencijalne informacije igraju ključnu ulogu.

5. **Konvolucione neuronske mreže (CNN)**

CNN su prvobitno razvijene za obradu slike, ali su uspešno primenjene i u NLP zadacima. U tekstualnoj klasifikaciji, CNN može da identifikuje lokalne obrasce u podacima, kao što su fraze ili n-grami, koji su značajni za klasifikaciju. CNN koristi konvolucione slojeve za ekstrakciju karakteristika i često je efikasniji od tradicionalnih RNN-ova u pogledu brzine treniranja.

**Analiza i poređenje rezultata**

**Logistička regresija** ostvarila je ukupnu tačnost od 90%. Za klasu 1 model pokazuje preciznost od 92% i recall od 95%, što ukazuje na visoku sposobnost detekcije pozitivnih primera. Za klasu 0, preciznost iznosi 81%, a recall 72%, što sugeriše prisustvo lažno pozitivnih predikcija. Ovaj model je pogodan za zadatke gde je prioritet detekcija klase 1, a balans između klasa nije ključan.

**SVM (Support Vector Machine)** model takođe postiže ukupnu tačnost od 90%. Za klasu 0, recall je 75%, dok je preciznost za klasu 1 93% i recall 95%. Ovaj balans performansi čini SVM model stabilnijim i pouzdanijim u scenarijima gde je važno ravnomerno performiranje za obe klase.

**Naive Bayes** model ima ukupnu tačnost od 84%. Za klasu 1, recall je impresivnih 98%, što pokazuje visoku sposobnost detekcije pozitivnih primera. Međutim, za klasu 0, recall je samo 37%, uz visoku preciznost od 87%. Ovaj model je idealan kada je ključno minimizirati propuštanje pozitivnih primera, iako se žrtvuje performansa za klasu 0.

**LSTM (Long Short-Term Memory)** model dostigao je ukupnu tačnost od 89%. Za klasu 1, preciznost je 92%, a recall 94%, što pokazuje dobru detekciju pozitivnih primera. Za klasu 0, preciznost je 78%, a recall 72%, što može biti problematično u scenarijima gde je važno prepoznati sve negativne primere. Iako LSTM modeli mogu uhvatiti kompleksne obrasce u podacima, njihova složenost i visoki zahtevi za resursima čine ih manje efikasnim za jednostavnije zadatke.

**CNN (Convolutional Neural Network)** model ostvario je ukupnu tačnost od 87%. Za klasu 1, preciznost je 90% i recall 94%, dok za klasu 0 preciznost iznosi 75%, a recall 64%. Ovaj slabiji balans između klasa čini CNN model manje pogodnim za primene gde je jednaka tačnost za sve klase ključna. Iako CNN može prepoznati kompleksne obrasce, njegova efikasnost u detekciji klase 0 je ograničena.

SVM model se izdvaja kao najstabilniji sa dobrim balansom performansi za obe klase, postigavši ukupnu tačnost od 90% i visok recall za klasu 0. Logistička regresija takođe pruža solidne rezultate sa istom tačnošću, ali sa nešto slabijim balansom između klasa. Naive Bayes je optimalan izbor kada je ključan visok recall za klasu 1, iako se performansa za klasu 0 smanjuje. LSTM i CNN modeli su prikladni za zadatke koji zahtevaju prepoznavanje kompleksnih obrazaca, ali zahtevaju dodatnu optimizaciju kako bi poboljšali balans između

klasa i generalizaciju. Konačni izbor modela zavisi od specifičnih zahteva zadatka, posebno u pogledu prioriteta između preciznosti i recall-a za svaku klasu.

### 5.4. Metode za augmentaciju podataka

U cilju poboljšanja performansi modela i povećanja raznolikosti skupa podataka, definisane su sledeće metode augmentacije tekstualnih podataka:

1. **Simulacija pravopisnih grešaka (`simulate_spelling_errors`)**: Umeće pravopisne greške u tekst sa određenom verovatnoćom kako bi model bio otporniji na greške u pisanju.
2. **Simulacija grešaka u kucanju (`keyboard_augmenter`)**: Simulira tipične greške koje nastaju zbog blizine tastera na tastaturi, oponašajući ljudske greške pri kucanju.
3. **Simulacija OCR grešaka (`ocr_simulation`)**: Umeće greške karakteristične za optičko prepoznavanje karaktera, povećavajući robusnost modela na takve nesavršenosti.
4. **Nasumična augmentacija karaktera (`random_character_augmentation`)**: Uključuje brisanje, zamenu, permutaciju ili umetanje karaktera u tekst kako bi se generisale različite varijante.
5. **Zamena sinonimima (`synonym_replacement`)**: Zamenjuje određeni broj reči njihovim sinonimima, obogaćujući vokabular i uvodeći semantičku raznolikost.
6. **Zamena antonimima (`antonym_replacement`)**: Zamenjuje reči njihovim antonimima, što testira sposobnost modela da razume promene u značenju.
7. **Nasumično umetanje reči (`random_insertion`)**: Umeće sinonime nasumično u tekst, povećavajući dužinu i složenost rečenica.
8. **Nasumična zamena reči (`random_swap`)**: Menja mesta dvema rečima u tekstu, stvarajući sintaksičke varijacije.
9. **Nasumično brisanje reči (`random_deletion`)**: Briše reči iz teksta sa određenom verovatnoćom, pomažući modelu da se nosi sa nepotpunim informacijama.
10. **Podela reči (`split_augmentation`)**: Deli reči na manje delove, simulirajući greške u pisanju ili kucanju.
11. **Augmentacija pravopisnim greškama (`spelling_augmentation`)**: Umeće pravopisne greške direktno u reči, povećavajući robusnost modela na pravopisne varijacije.
12. **Back-Translation (`back_translation`)**: Prevođenjem teksta na drugi jezik i nazad dobija se parafraziran tekst koji zadržava originalno značenje.
13. **Nasumična augmentacija rečenica (`random_sentence_augmentation`)**: Menja redosled rečenica u tekstu, uvodeći varijacije u strukturi i toku misli.

Radi jednostavnosti, samo neke od ovih metoda će biti primenjene u svrhu augmentacije seta podataka, čime će se povećati njegova raznolikost i poboljšati generalizacija modela za detekciju govora mržnje i uvredljivog jezika.

## 5.5. Primena modela nad augmentiranim dataset-ovima

Nakon primene metoda za augmentaciju podataka na klasu 0 (nije govor mržnje), kreirana su dva augmentirana skupa podataka kako bi se dodatno poboljšala raznovrsnost i balansiranost skupa. Na svakom od ovih skupa podataka primenjene su iste mašinske metode kako bi se analizirali efekti različitih tehnika augmentacije na performanse modela. Sledeća analiza obuhvata rezultate dobijene na oba augmentirana skupa podataka.

### *4.5.1. Primena modela nad prvim augmentiranim dataset-om*

Prvi augmentirani dataset je kreiran korišćenjem sledećih metoda: simulacija pravopisnih grešaka (simulate_spelling_errors), zamena sinonima (synonym_replacement), nasumična augmentacija rečenica (random_sentence_augmentation) i splitovanje (split_augmentation). Nakon augmentacije, skup podataka je balansiran sa 19,189 instanci klase 0 i 19,190 instanci klase 1.

Augmentacija je doprinela povećanju raznovrsnosti podataka, što se ogleda u poboljšanoj tačnosti, boljoj ravnoteži između klasa i poboljšanim metrikama performansi u poređenju sa prethodnim rezultatima bez augmentacije.

U setu podataka augmentiranom korišćenjem tradicionalnih metoda, logistička regresija postiže ukupnu tačnost od 91%, što predstavlja blago poboljšanje u odnosu na prethodnih 90%. Preciznost za klasu 0 i klasu 1 ostaje na visokom nivou (90% i 92%), dok se recall za klasu 0 povećao sa 72% na 92%, a za klasu 1 se smanjio sa 95% na 89%. Ovo ukazuje na značajno poboljšanje u detekciji klase 0, uz blago smanjenje u prepoznavanju klase 1. Ukupan balans između klasa je sada izjednačeniji, što čini model pogodnijim za zadatke gde je važno održati ravnotežu između detekcije obe klase.

SVM model pokazuje značajno poboljšanje sa ukupnom tačnošću od 93%, u poređenju sa prethodnih 90%. Preciznost za obe klase je blago povećana (91% na 95% za klasu 1), dok se recall za klasu 0 povećao sa 75% na impresivnih 95%. Ovo rezultira odličnim balansom između klasa, smanjujući broj lažno pozitivnih i lažno negativnih predikcija. SVM nastavlja da se izdvaja kao najstabilniji i najpouzdaniji model, sa poboljšanim performansama koje čine ovaj model izuzetno pogodnim za zadatke gde je ravnoteža između klasa ključna.

Naive Bayes model u novom skupu podataka postiže ukupnu tačnost od 86%, što predstavlja blago povećanje u odnosu na prethodnih 84%. Preciznost za klasu 0 ostaje visoka (90%), dok se preciznost za klasu 1 smanjuje sa 84% na 82%. Recall za klasu 0 se poboljšava sa 37% na 81%, što značajno smanjuje broj lažno negativnih predikcija za ovu klasu. Istovremeno, recall za klasu 1 ostaje visok (90%), što ukazuje na sposobnost modela da detektuje skoro sve pozitivne primere. Ova poboljšanja čine Naive Bayes model efikasnijim u balansiranju između klasa, iako preciznost za klasu 1 ostaje nešto niža u poređenju sa SVM i Logističkom regresijom.

LSTM model u ovom novom eksperimentu postiže ukupnu tačnost od 93%, što je značajno poboljšanje u odnosu na prethodnih 89%. Preciznost i recall za obe klase su se poboljšali na 93% i 92% za klasu 0, te 92% i 93% za klasu 1. Ovo pokazuje da je LSTM model postigao bolji balans između detekcije obe klase, smanjujući broj lažno pozitivnih i lažno negativnih predikcija. Ova poboljšanja ukazuju na bolju sposobnost modela da uhvati kompleksne obrasce u podacima, uz efikasniju generalizaciju.

CNN model u novom skupu podataka postiže ukupnu tačnost od 91%, što je značajno poboljšanje u odnosu na prethodnih 87%. Preciznost za klasu 0 i klasu 1 povećana je na 90% i 92%, dok se recall za klasu 0 i klasu 1 povećava na 92% i 90%. Ovo rezultira boljim balansom između klasa, smanjujući broj lažno pozitivnih i lažno negativnih predikcija. Ova poboljšanja čine CNN model efikasnijim i pogodnijim za primene gde je važno održati visok nivo tačnosti za obe klase.

Upoređujući nove rezultate sa prethodnim, većina modela je pokazala poboljšanje u ključnim metrikama. SVM ostaje najstabilniji i najpouzdaniji model, sa značajnim poboljšanjima u tačnosti i balansu između klasa. Logistička regresija je unapređena, naročito u detekciji klase 0, čineći je pogodnijom za zadatke gde je potrebna bolja ravnoteža između klasa. Naive Bayes je značajno poboljšao recall za klasu 0, čime je postao efikasniji u detekciji negativnih primera, iako je preciznost za klasu 1 malo opala. LSTM i CNN modeli su takođe pokazali značajan napredak, sa poboljšanjem tačnosti i boljim balansom između klasa, čineći ih efikasnijim za složenije zadatke.

Ukupno gledano, novi rezultati ukazuju na unapređenja u performansama svih modela, sa najznačajnijim poboljšanjima kod SVM, LSTM i CNN modela. Ova poboljšanja čine modele još pogodnijim za različite scenarije klasifikacije, zavisno od specifičnih zahteva zadatka, posebno u pogledu prioriteta između preciznosti i recall-a za svaku klasu.

### *4.5.2. Primena modela nad drugim augmentiranim dataset-om*

Drugi augmentirani dataset je kreiran korišćenjem naprednijih metoda augmentacije podataka putem **TextAttack** biblioteke, uključujući **CLAREAugmenter** i **EmbeddingAugmenter**. Clare Augmenter kombinuje pravila i kontekstualne informacije kako bi izvršio složenije transformacije teksta, koristeći model distilroberta-base. Takođe, dodatno je izvršena augmentacija korišćenjem naprednih jezičkih modela kao što su GPT-2 i parafraziranje pomoću T5. Ove metode omogućavaju generisanje novih tekstualnih instanci kroz generativne modele, čime se dodatno povećava raznolikost i kvalitet augmentiranih podataka.

Ove metode omogućavaju kontekstualno prilagođavanje i raznovrsnije transformacije tekstualnih podataka, što doprinosi većoj raznolikosti i kvalitetu augmentisanih podataka.

Nakon augmentacije, skup podataka sadrži 42783 instance, pri čemu je 23,593 instanci klase 0 i 19,190 instanci klase 1.

Analiza rezultata ukazuje na značajno poboljšanje performansi modela usled primene naprednih metoda augmentacije. Ove tehnike su omogućile modelima da efikasnije generalizuju i preciznije klasifikuju podatke, što se ogleda u povećanju tačnosti, ravnoteži metrika između klasa, i smanjenju grešaka u predikcijama.

Logistička regresija na trećem datasetu postiže ukupnu tačnost od 91%, sa visokim preciznostima i recall-om za obe klase. Preciznost za klasu 0 je 90%, a za klasu 1 je 92%. Recall za klasu 0 je 93%, dok je za klasu 1 88%. Ovo pokazuje uravnoteženu detekciju obe klase, sa blagim padom recall-a za klasu 1 u odnosu na prethodni dataset.

SVM model na trećem datasetu postiže ukupnu tačnost od 93%, što je značajno poboljšanje u odnosu na originalni dataset (90%) Preciznost za klasu 0 je 92%, a za klasu 1 95%. Recall za klasu 0 je 96%, dok je za klasu 1 90%. Ovaj model pokazuje izuzetno visok balans između klasa, sa minimalnim brojem lažno pozitivnih i lažno negativnih predikcija.

Naive Bayes model na trećem datasetu postiže ukupnu tačnost od 87%, što je poboljšanje u odnosu na originalni dataset (84%) i tradicionalno augmentirani dataset (86%). Preciznost za klasu 0 je 90%, dok je za klasu 1 83%. Recall za klasu 0 je 85%, a za klasu 1 89%. Ovo pokazuje značajno poboljšanje u detekciji klase 0, uz blagi pad preciznosti za klasu 1 u odnosu na drugi dataset.

LSTM model na ovom datasetu postiže ukupnu tačnost od 93%, sa preciznostima od 92% za klasu 0 i 94% za klasu 1. Recall za klasu 0 je 95%, dok je za klasu 1 90%. Ovo pokazuje visok balans između detekcije obe klase, sa značajnim poboljšanjem u odnosu na originalni (89%) dataset.

CNN model na trećem datasetu postiže ukupnu tačnost od 92%, sa preciznostima od 91% za klasu 0 i 94% za klasu 1. Recall za klasu 0 je 95%, dok je za klasu 1 89%. Ovo pokazuje poboljšanje u odnosu na originalni dataset (87%) i tradicionalno augmentirani dataset (91%), sa boljim balansom između klasa.

Ovi rezultati pokazuju da su naprednije metode augmentacije, doprinele boljoj generalizaciji modela i višim performansama u detekciji govora mržnje. Klasični modeli, posebno SVM i Logistička regresija, nastavljaju da pokazuju odlične performanse, dok su duboke neuronske mreže takođe pokazale značajno poboljšanje, što sugeriše da dodatna augmentacija podataka može doprineti njihovoj superiornosti u složenijim zadacima klasifikacije.

## 5.6. Tabelarni i vizuelni prikaz i analiza rezultata

**Logistička regresija:**

| Dataset | Tačnost (%) | F1 score – klasa 0 | F1 score – klasa 1 |
|---|---|---|---|
| Osnovni | 90 | 76 | 93 |
| Prvi augmentirani | 91 | 91 | 91 |
| Drugi augmentirani | 91 | 92 | 90 |

**SVM:**

| Dataset | Tačnost (%) | F1 score – klasa 0 | F1 score – klasa 1 |
|---|---|---|---|
| Osnovni | 90 | 78 | 94 |
| Prvi augmentirani | 93 | 93 | 92 |
| Drugi augmentirani | 93 | 94 | 93 |

**Naive Bayes:**

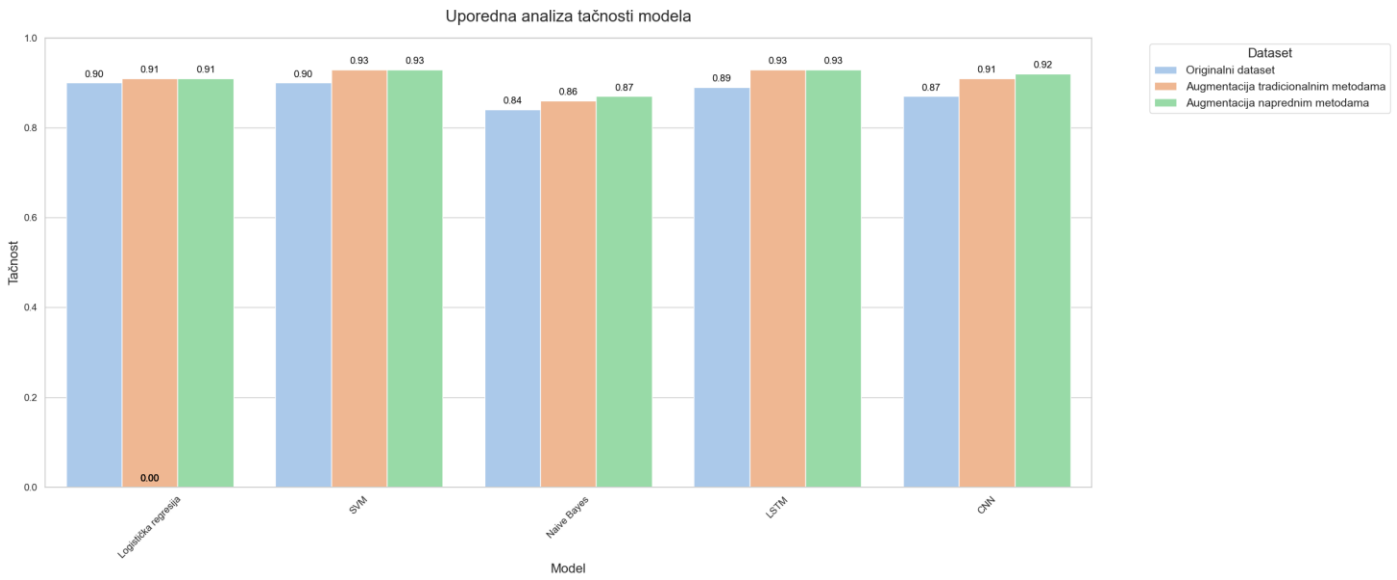
| Dataset | Tačnost (%) | F1 score – klasa 0 | F1 score – klasa 1 |
|---|---|---|---|
| Osnovni | 84 | 52 | 91 |
| Prvi augmentirani | 86 | 85 | 86 |
| Drugi augmentirani | 87 | 88 | 86 |

**LSTM:**

| Dataset | Tačnost (%) | F1 score – klasa 0 | F1 score – klasa 1 |
|---|---|---|---|
| Osnovni | 89 | 75 | 93 |
| Prvi augmentirani | 93 | 93 | 92 |
| Drugi augmentirani | 93 | 94 | 92 |

**CNN:**

| Dataset | Tačnost (%) | F1 score – klasa 0 | F1 score – klasa 1 |
|---|---|---|---|
| Osnovni | 87 | 69 | 92 |
| Prvi augmentirani | 91 | 91 | 91 |
| Drugi augmentirani | 92 | 93 | 91 |




Augmentacija podataka značajno poboljšava performanse modela u odnosu na originalni dataset. Napredne metode augmentacije daju blago bolje rezultate u poređenju sa tradicionalnim metodama, posebno kod modela kao što su CNN (92% vs. 91%) i Naive Bayes (86% vs. 87%). Originalni dataset pokazuje niže tačnosti, naročito kod Naive Bayes (84%) i CNN (87%), dok napredne metode omogućavaju veću stabilnost i maksimalne tačnosti od 93% kod LSTM i SVM modela. Sveukupno, napredna augmentacija se pokazuje kao najefikasnija za poboljšanje modela.

## 6. Zaključak

Ovaj rad je pružio sveobuhvatan pregled metoda augmentacije tekstualnih podataka, sa posebnim fokusom na detekciju govora mržnje kao ključnog izazova u savremenim NLP primenama. Istaknuta je važnost kvalitetne i raznovrsne obrade podataka, koja predstavlja osnovu za izgradnju robusnih i preciznih modela mašinskog učenja.

Detaljno su opisani koraci predobrade teksta, poput čišćenja, normalizacije, tokenizacije i lematizacije, koji osiguravaju doslednost i smanjuju šum u podacima. Pregled metoda augmentacije podeljen je u dve glavne kategorije – Data Space i Feature Space – uz analizu jednostavnih i naprednih tehnika, od sinonimnih zamena i back-translationa, do generativnih metoda zasnovanih na modelima kao što su GPT, BERT i T5.

Praktični deo rada pokazao je kako augmentacija podataka direktno utiče na poboljšanje performansi različitih modela mašinskog učenja. Napredne metode augmentacije, poput parafraziranja pomoću generativnih modela i contextual embeddings, dale su bolje rezultate u poređenju sa tradicionalnim pristupima. Modeli poput SVM i LSTM pokazali su visoku tačnost i stabilnost, dok su klasični modeli, poput Naive Bayes i Logističke regresije, ostvarili značajna poboljšanja uz primenu augmentacije.

Kroz ovaj rad jasno je pokazano da pravilno primenjena augmentacija tekstualnih podataka unapređuje sposobnost modela da generalizuju i prepoznaju kompleksne obrasce u tekstu. Kombinovanjem različitih tehnika augmentacije, moguće je značajno povećati otpornost i efikasnost modela u detekciji govora mržnje, pružajući time dragocene smernice za primenu u realnim NLP zadacima.

## 7. Reference


Agrawal, R. (2021, June 14). *Must Known Techniques for text preprocessing in NLP*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/must-known-techniques-for-text-preprocessing-in-nlp/

Arslan, E. (2024, June 25). *Natural language Processing: Deep dive into text preprocessing and tokenization— step 2*. Medium. https://medium.com/@erhan_arslan/natural-language-processing-deep-dive-into-text-preprocessing-and-tokenization-step-2-b5dcf9520266

Aydin, A. (2023, October 4). *1 — text preprocessing techniques for NLP*. Medium. https://ayselaydin.medium.com/1-text-preprocessing-techniques-for-nlp-37544483c007

Bismi, I. (2023, April 2). *Augmenting text using large language models: GPT-2, GPT-3, BERT*. Medium. https://medium.com/@iqra.bismi/augmenting-text-using-large-language-models-gpt-2-gpt-3-bert-b6ca8008d85d

Bolle, M. (2023, June 3). *Text augmentation in python with NLPAUG - Marc Bolle*. Medium. https://medium.com/@marc.bolle/text-augmentation-in-python-with-nlpaug-48c3eebacf46

Chiusano, F. (2022, April 4). *Two minutes NLP — A taxonomy of data augmentation for text classification*. NLPlanet. https://medium.com/nlplanet/two-minutes-nlp-a-taxonomy-of-data-augmentation-for-text-classification-52c96f332bad

Claude, C. (2018). Text data augmentation made simple by leveraging NLP Cloud APIs. In *arXiv [cs.CL]*. http://arxiv.org/abs/1812.04718

Dianqi, L., Yizhe, Z., Hao, P., Liqun, C., Chris, B., Ming-Ting, S., & Bill, D. (2020). Contextualized perturbation for textual adversarial attack. In *arXiv [cs.CL]*. http://arxiv.org/abs/2009.07502

*Getting started with text preprocessing*. (2019, August 20). Kaggle.com; Kaggle. https://www.kaggle.com/code/sudalairajkumar/getting-started-with-text-preprocessing

Li, B., Hou, Y., & Che, W. (2022). Data augmentation approaches in natural language processing: A survey. *AI Open*, *3*, 71–90. https://doi.org/10.1016/j.aiopen.2022.03.001

Ma, E. (2019, April 20). *Data Augmentation library for text*. Towards Data Science. https://towardsdatascience.com/data-augmentation-library-for-text-9661736b13ff

Markus, B., Marc-André, K., & Christian, R. (2021). A Survey on Data Augmentation for Text Classification. In *arXiv [cs.CL]*. http://arxiv.org/abs/2107.03158

Pellicer, L. F. A. O., Ferreira, T. M., & Costa, A. H. R. (2023). Data augmentation techniques in natural language processing. *Applied Soft Computing*, *132*(109803), 109803. https://doi.org/10.1016/j.asoc.2022.109803

*Text augmentation techniques in NLP*. (2023, April 17). GeeksforGeeks. https://www.geeksforgeeks.org/text-augmentation-techniques-in-nlp/

*Text to text Transfer Transformer in Data Augmentation*. (2021, January 20). GeeksforGeeks. https://www.geeksforgeeks.org/text-to-text-transfer-transformer-in-data-augmentation/

Van Otten, N. (2023, January 25). *How to use text normalization techniques in NLP with python [9 ways]*. Spot Intelligence. https://spotintelligence.com/2023/01/25/text-normalization-techniques-nlp/

*What are the best practices for using generative adversarial networks in data augmentation?* (n.d.). Linkedin.com. Retrieved December 8, 2024, from https://www.linkedin.com/advice/0/what-best-practices-using-generative-adversarial-1awbf