УНИВЕРЗИТЕТ У НИШУ
ЕЛЕКТРОНСКИ ФАКУЛТЕТ

Metode za augmentaciju tekstualnih podatka

Projekat
Predmet: Prikupljanje i predobrada podataka za masinsko ucenje

Student:

Aleksa Milic, broj indeksa 1610

Mentor:

Aleksandar Stanimirovic

Nis, 2024. godina

Sadrzaj

Ниш,2024. година

1. Augmentacija teksta: Revolucija u obradi prirodnog jezika

U oblasti obrade prirodnog jezika (NLP) koja se brzo razvija, kvalitet i količina podataka igraju ključnu ulogu u određivanju uspeha modela mašinskog učenja. Dok se trudimo da razvijemo sofisticiranije algoritme sposobne da razumeju i generišu tekst nalik ljudskom, često se suočavamo sa značajnim izazovom: nedostatkom visokokvalitetnih, označenih podataka. Upravo tu augmentacija teksta izranja kao revolucionarna tehnika, nudeći rešenje za večiti problem nedostatka podataka.

Dilema podataka u NLP-u

Zamislite da imate zadatak da razvijete model za analizu sentimenta za specifičnu industriju. Brzo shvatate da, iako imate pristup ogromnim količinama tekstualnih podataka, samo je mali deo označen, što ozbiljno ograničava vašu sposobnost da trenirate robustan model. Ovaj scenario nije neuobičajen u svetu NLP-a, gde prikupljanje i označavanje podataka može biti vremenski zahtevno i skupo.

Posledice nedovoljnih podataka su dalekosežne:

1. **Ograničene performanse modela**: Modeli trenirani na malim skupovima podataka često se bore da dobro generalizuju na nove, neviđene podatke.
2. **Prekomerno prilagođavanje (overfitting)**: Sa ograničenim primerima, modeli mogu memorisati trening podatke umesto da uče generalizovane obrasce.
3. **Pristrasnost**: Mali skupovi podataka možda ne predstavljaju punu raznolikost upotrebe jezika, što dovodi do pristrasnih izlaza modela.

Ulazak augmentacije teksta

Augmentacija teksta nudi moćno rešenje za ove izazove. Veštačkim proširivanjem našeg skupa podataka kroz različite tehnike, možemo značajno poboljšati performanse i sposobnosti generalizacije naših NLP modela. Hajde da istražimo zašto je augmentacija teksta postala nezamenljiv alat u kompletu NLP praktičara:

1. Povećanje veličine i raznolikosti skupa podataka

Augmentacija teksta nam omogućava da generišemo nove, sintetičke podatke iz našeg postojećeg skupa podataka. Ovo ne samo da povećava obim podataka za trening, već i uvodi vrednu raznolikost. Na primer, primenom zamene sinonimima, možemo kreirati višestruke varijacije jedne rečenice, svaka prenoseći isto značenje ali sa malo drugačijim rečima.

2. Rešavanje neravnoteže podataka

U zadacima kao što je klasifikacija namera, gde određene klase mogu biti nedovoljno zastupljene, tehnike augmentacije se mogu selektivno primeniti za uravnoteženje skupa podataka. Ovo osigurava da naši modeli ne razviju pristrasnost prema previše zastupljenim klasama.

3. Poboljšanje robusnosti modela

Izlažući naše modele širem spektru lingvističkih obrazaca i izraza tokom treninga, augmentacija teksta pomaže u izgradnji robusnijih modela koji mogu bolje da se nose sa nijansama i složenošću prirodnog jezika.

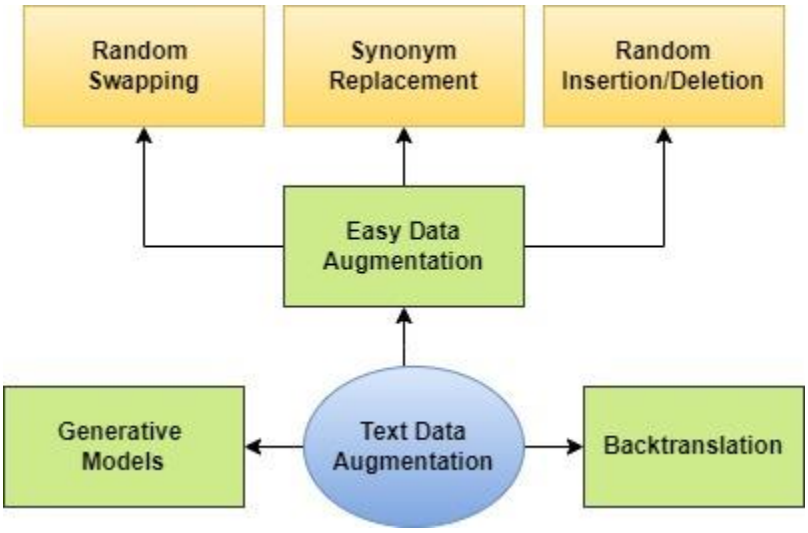
4. **Ekonomično rešenje**
   Dok je ručno označavanje podataka skupo i vremenski zahtevno, augmentacija teksta nudi ekonomično alternativno rešenje za proširenje našeg skupa podataka. Omogućava nam da maksimalno iskoristimo naše postojeće označene podatke, potencijalno štedeći značajne resurse u prikupljanju i označavanju podataka.

Arsenal augmentacije

Augmentacija teksta obuhvata širok spektar tehnika, svaka sa svojim prednostima i primenama:

1. **Zamena sinonimima**: Zamena reči njihovim sinonimima za kreiranje varijacija uz očuvanje značenja.
2. **Povratni prevod**: Prevođenje teksta na drugi jezik i nazad na originalni, uvođenje prirodnih varijacija.
3. **Generisanje teksta pomoću jezičkih modela**: Korišćenje naprednih jezičkih modela poput GPT-a za generisanje kontekstualno relevantnog novog teksta.
4. **Umetanje i brisanje reči**: Nasumično dodavanje ili uklanjanje reči za simulaciju prirodnih jezičkih varijacija.
5. **Mešanje rečenica**: Preuređivanje rečenica u dokumentu za kreiranje novih primera uz održavanje opšteg konteksta.

Priložena slika pruža sveobuhvatan pregled različitih pristupa augmentaciji teksta, od jednostavnih manipulacija na nivou reči do složenih generativnih tehnika.



Slika 1. Različite tehnike augmentacije tekstualnih podataka.

2. Preprocesiranje podataka u obradi prirodnog jezika

Preprocesiranje podataka je suštinski korak u obradi prirodnog jezika (NLP) i analizi teksta, jer omogućava da se sirovi tekstualni podaci transformišu u strukturu pogodnu za analizu i primenu algoritama mašinskog učenja. U svetu gde se svakodnevno generiše ogromna količina tekstualnih informacija, efikasno preprocesiranje je neophodno kako bi se iz te mase podataka izvukle relevantne informacije koje su korisne za donošenje odluka, predviđanja ili automatizaciju procesa.

Sirovi tekstualni podaci, kao što su komentari na društvenim mrežama, članci iz novina, e-mailovi i drugi oblici tekstualnog sadržaja, često sadrže šumove, greške, višeznačne izraze i druge nepravilnosti koje mogu ometati rad NLP modela. Preprocesiranje je faza koja obuhvata niz metoda i tehnika dizajniranih za čišćenje i pripremu tih podataka, čineći ih spremnim za dalju analizu.

Kvalitetno preprocesiranje ne samo da poboljšava tačnost i efikasnost NLP modela, već može značajno smanjiti kompleksnost problema, omogućavajući modelima da se fokusiraju na suštinske informacije. Na primer, eliminacijom nevažnih reči, normalizacijom teksta ili uklanjanjem šumova, modeli mogu bolje da identifikuju obrasce i odnose unutar teksta, što direktno utiče na performanse modela.

U ovom odeljku, detaljno ćemo istražiti pet ključnih tehnika preprocesiranja podataka: čišćenje teksta, normalizaciju, tokenizaciju, lematizaciju i stemovanje, kao i uklanjanje stop-reči. Svaka od ovih tehnika ima specifičnu ulogu i doprinosi stvaranju čistih, doslednih i analitički korisnih tekstualnih podataka, koji su osnov za dalju obradu i analizu u NLP zadacima. Razumevanje i pravilna primena ovih tehnika ključni su za izgradnju moćnih NLP rešenja koja mogu efikasno obraditi i analizirati prirodni jezik.

2.1 Preprocesiranje podataka u obradi prirodnog jezika

Obrada podataka se tiče osnovnog čišćenja ulaznih tekstualnih podataka i uključuje neke od sledećih tehnika:

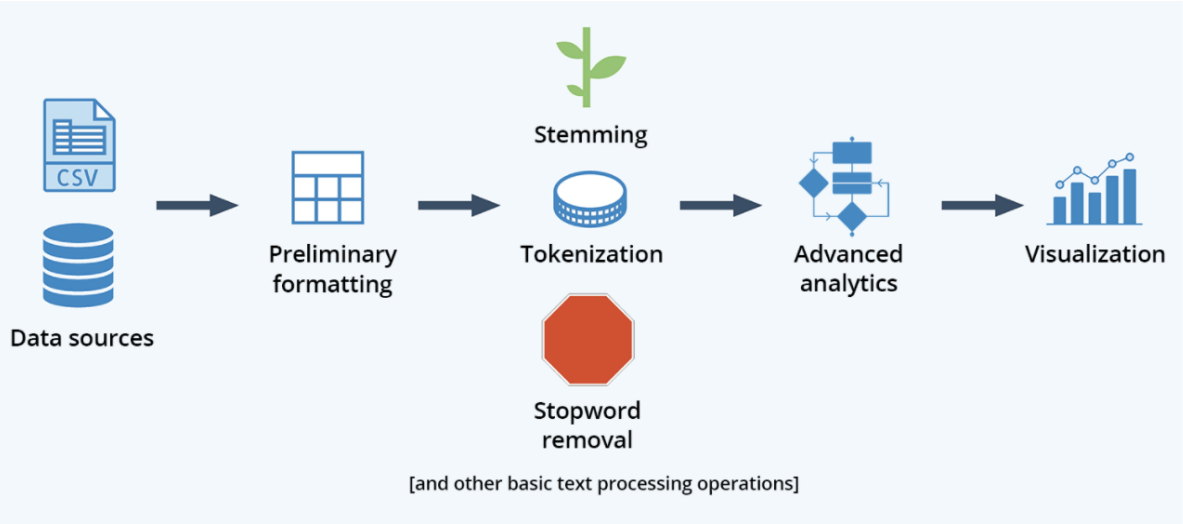
1. Čišćenje teksta

2. Normalizacija

3. Tokenizacija

4. Lematizacija i stevanje

5. Uklanjanje stop-reči



Slika 2. Opsti process pripreme teksta

2.1.1 Čišćenje teksta

Čišćenje teksta je prvi i često najkritičniji korak u preprocesiranju podataka. Ovaj proces podrazumeva uklanjanje ili modifikaciju delova teksta koji nisu relevantni za analizu ili mogu negativno uticati na performanse modela.

Ključni aspekti čišćenja teksta uključuju:

1. **Uklanjanje HTML oznaka i specijalnih karaktera**:
   - Mnogi tekstovi, posebno oni prikupljeni sa veba, mogu sadržati HTML oznake ili specijalne karaktere koji nisu relevantni za semantičku analizu.
   - Primer: Transformacija Ovo je primer teksta. u Ovo je primer teksta.

2. **Uklanjanje ili zamena brojeva**:
   - U zavisnosti od zadatka, brojevi mogu biti irelevantni ili mogu zahtevati standardizaciju.
   - Primer: Zamena Imam 42 godine sa Imam NUM godine

3. **Uklanjanje ili zamena URL-ova i e-mail adresa**:
   - URL-ovi i e-mail adrese često nisu relevantni za semantičku analizu i mogu se zameniti generičkim tokenima.
   - Primer: Zamena Posetite www.example.com sa Posetite URL

4. **Uklanjanje viška belina i novih redova**:
   - Višak belina i novi redovi mogu otežati obradu teksta i treba ih normalizovati.
   - Primer: Transformacija Ovo je primer teksta u Ovo je primer teksta

5. **Uklanjanje ili zamena emotikona i emoji-a**:
   - U zavisnosti od zadatka, emotikoni i emoji mogu biti zamenjeni opisnim tekstom ili potpuno uklonjeni.
   - Primer: Zamena Danas je lep dan :) sa Danas je lep dan HAPPY_FACE

6. **Ispravljanje čestih pravopisnih grešaka**:
    - Automatsko ispravljanje čestih grešaka može poboljšati kvalitet podataka.
    - Primer: Ispravljanje Ja neznma u Ja ne znam

Značaj čišćenja teksta:

- **Poboljšana konzistentnost**: Čišćenje teksta osigurava da su podaci konzistentni, što je ključno za pouzdanu analizu.
- **Smanjenje šuma**: Uklanjanjem irelevantnih informacija, modeli se mogu fokusirati na suštinski važne aspekte teksta.
- **Povećana efikasnost**: Čist tekst je lakši za obradu, što može ubrzati trening modela i smanjiti računske zahteve.

2.1.2 Normalizacija

Normalizacija je proces standardizacije teksta, čime se smanjuju varijacije koje mogu otežati analizu. Cilj normalizacije je predstaviti slične oblike reči ili fraza na konzistentan način.
Ključne tehnike normalizacije uključuju:

1. **Konverzija u mala slova**:
    - Pretvaranje svih karaktera u mala slova pomaže u smanjenju dimenzionalnosti vokabulara.
    - Primer: Transformacija Ovo Je PRIMER u ovo je primer

2. **Uklanjanje akcenata**:
    - U jezicima koji koriste akcente, njihovo uklanjanje može pomoći u standardizaciji teksta.
    - Primer: Transformacija café u cafe

3. **Standardizacija formata datuma i vremena**:
    - Različiti formati datuma i vremena mogu se konvertovati u standardni format.
    - Primer: Transformacija 01/02/2023 i 1. februar 2023. u 2023-02-01

4. **Ekspanzija skraćenica**:
    - Proširivanje čestih skraćenica može poboljšati razumevanje teksta.
    - Primer: Transformacija Dr. u Doktor

5. **Standardizacija interpunkcije**:
    - Konzistentna upotreba interpunkcije može olakšati dalju obradu.
    - Primer: Transformacija Zdravo!!! u Zdravo.

Značaj normalizacije:

- **Smanjenje dimenzionalnosti**: Normalizacija smanjuje broj jedinstvenih tokena u korpusu, što može poboljšati performanse modela.
- **Poboljšana generalizacija**: Standardizovani tekst omogućava modelima da bolje generalizuju na nove primere.
- **Konzistentnost u analizi**: Normalizacija osigurava da slični koncepti budu tretirani na isti način tokom analize.

2.1.3 Токенизација

Tokenizacija je proces razbijanja teksta na manje jedinice, obično reči ili karaktere. Ovo je fundamentalan korak u većini NLP zadataka, jer pretvara neprekidni tekst u diskretne jedinice koje se mogu analizirati.

Ključni aspekti tokenizacije:

1. **Tokenizacija na nivou reči**:
   - Razbijanje teksta na pojedinačne reči.
   - Primer: Ovo je primer rečenice. → ['Ovo', 'je', 'primer', 'rečenice', '.']

2. **Tokenizacija na nivou karaktera**:
   - Razbijanje teksta na pojedinačne karaktere.
   - Primer: Zdravo → ['Z', 'd', 'r', 'a', 'v', 'o']

3. **Tokenizacija na nivou podreči (subword tokenization)**:
   - Razbijanje reči na manje jedinice, što je posebno korisno za morfološki bogate jezike ili za rukovanje nepoznatim rečima.
   - Primer: najlepši → ['naj', 'lep', 'ši']

4. **Rukovanje sa interpunkcijom**:
   - Odlučivanje kako tretirati interpunkciju (kao posebne tokene ili spojeno sa rečima).
   - Primer: Zdravo, svete! → ['Zdravo', ',', 'svete', '!'] ili ['Zdravo,', 'svete!']

5. **Rukovanje sa višerečnim izrazima**:
   - Identifikacija i očuvanje višerečnih izraza kao jednog tokena.
   - Primer: Novi Sad → ['Novi_Sad'] umesto ['Novi', 'Sad']

Značaj tokenizacije:

- **Osnova za vektorsku reprezentaciju**: Tokenizacija je preduslov za pretvaranje teksta u numeričke vektore koje modeli mašinskog učenja mogu obraditi.
- **Fleksibilnost u analizi**: Različiti nivoi tokenizacije omogućavaju različite pristupe analizi teksta.
- **Prilagođavanje jezičkim specifičnostima**: Pravilna tokenizacija uzima u obzir specifičnosti jezika, poput složenica u nemačkom ili aglutinativne prirode turskog jezika.

2.1.4 Lematizacija i stemovanje

Lematizacija i stemovanje su tehnike koje se koriste za svođenje reči na njihove osnovne oblike. Iako imaju sličan cilj, ove tehnike se razlikuju u pristupu i rezultatima.

Stemovanje:
Stemovanje je proces uklanjanja afiksa (prefiksa i sufiksa) reči kako bi se dobio koren reči ili "stem". Ova tehnika je brža i jednostavnija, ali može rezultirati nepostojećim rečima.

Karakteristike stemovanja:

- **Brzina**: Stemovanje je računski efikasno i brzo.
- **Jednostavnost**: Koristi set pravila za uklanjanje afiksa.
- **Neprec

iznost**: Može proizvesti nepravilne ili nepostojeće korene reči.

Primeri stemovanja:

- trčanje → trč
- trčati → trč
- trčim → trč

Lematizacija:

Lematizacija je sofisticiraniji proces koji svodi reči na njihove rečničke (kanonske) oblike, poznate kao leme. Ova tehnika uzima u obzir morfologiju reči i kontekst u kojem se pojavljuje.

Karakteristike lematizacije:

- **Preciznost**: Proizvodi stvarne, rečničke oblike reči.
- **Kontekstualna svest**: Uzima u obzir kontekst reči za pravilnu lematizaciju.
- **Računska zahtevnost**: Zahtevnija je u pogledu resursa i vremena obrade.

Primeri lematizacije:

- trčanje → trčati
- bolja → dobar
- bila → biti

Značaj lematizacije i stemovanja:

- **Smanjenje dimenzionalnosti**: Smanjenjem broja jedinstvenih reči u korpusu, ove tehnike pomažu u smanjenju dimenzionalnosti vektorskog prostora.
- **Poboljšana analiza teksta**: Omogućavaju povezivanje različitih oblika iste reči, što je korisno za zadatke poput pretraživanja informacija i analize sentimenta.
- **Rukovanje sa retkim rečima**: Pomažu u rešavanju problema retkih reči smanjenjem varijacija reči.

2.1.5 Uklanjanje stop-reči

Uklanjanje stop-reči je proces eliminacije čestih reči koje obično ne nose značajnu semantičku vrednost u analizi teksta. Ove reči, poput veznika, predloga i članova, često se pojavljuju u tekstu ali retko doprinose njegovom značenju.

Ključni aspekti uklanjanja stop-reči:

1. **Definisanje liste stop-reči**:
    - Liste stop-reči mogu varirati u zavisnosti od jezika i specifičnog zadatka.
    - Primeri stop-reči u srpskom jeziku: i, u, na, je, da, se, za, koji, ne, su

2. **Prilagođavanje liste zadatku**:
    - Neke reči koje se generalno smatraju stop-rečima mogu biti značajne u određenim kontekstima.
    - Primer: U analizi sentimenta, reč "ne" može biti kritična i ne bi trebalo da bude uklonjena

3. **Balansiranje između smanjenja šuma i očuvanja značenja**:
    - Preagresivno uklanjanje stop-reči može dovesti do gubitka konteksta i značenja.

4. **Rukovanje sa višejezičnim tekstovima**:
    - Za višejezične korpuse, potrebno je koristiti odgovarajuće liste stop-reči za svaki jezik.

Primeri uklanjanja stop-reči:

- Pre: Ovo je primer rečenice koja sadrži nekoliko stop-reči.
- Posle: primer rečenice sadrži nekoliko stop-reči.

Značaj uklanjanja stop-reči:

- **Smanjenje dimenzionalnosti**: Uklanjanje čestih reči značajno smanjuje veličinu vokabulara i dimenzionalnost vektorskih reprezentacija.
- **Fokus na značajnim rečima**: Omogućava modelima da se fokusiraju na reči koje nose više semantičke informacije.
- **Poboljšana efikasnost**: Može ubrzati obradu i smanjiti zahteve za skladištenjem, posebno u velikim korpusima teksta.

3. Tehnike za augmentaciju teksta su:

Osnovne tehnike za augmentaciju koje ce biti detaljno opisane su:

- Augmentacija karaktera
- Augmentacija reči
- Augmentacija rečenica

3.1 Augmentacija karaktera u obradi prirodnog jezika

Augmentacija karaktera predstavlja skup tehnika koje se koriste u obradi prirodnog jezika (NLP) za modifikaciju tekstualnih podataka na nivou pojedinačnih znakova. Ove tehnike su ključne za poboljšanje robustnosti i generalizacije NLP modela, omogućavajući im da se bolje nose sa raznolikošću i nepravilnostima u realnim tekstualnim podacima.

Značaj augmentacije karaktera:

1. **Povećanje raznolikosti podataka**: Stvaranjem varijacija postojećeg teksta, modeli se izlažu širem spektru mogućih ulaza.
2. **Simulacija realnih scenarija**: Ove tehnike oponašaju stvarne greške i varijacije koje se javljaju u tekstu, pripremajući modele za rad sa nesavršenim podacima.
3. **Poboljšanje otpornosti na greške**: Modeli trenirani na augmentiranim podacima bolje se nose sa neočekivanim ulazima i greškama u tekstu.

Ključne tehnike augmentacije karaktera

1. Simulacija grešaka optičkog prepoznavanja karaktera (OCR)
OCR simulacija se fokusira na repliciranje grešaka koje nastaju pri digitalizaciji štampanih ili rukopisnih tekstova.
Primeri primene:

- Zamena slova 'o' sa '0' ili 'l' sa '1'
- Spajanje ili razdvajanje reči (npr. "zajedno" → "za jedno")
- Dodavanje mrlja ili šuma koji imitiraju oštećenja na papiru

Benefit: Priprema modela za rad sa tekstom ekstrahovanim iz slika ili skeniranih dokumenata.

2. Emulacija grešaka pri kucanju
Ova tehnika simulira uobičajene greške koje nastaju prilikom unosa teksta putem tastature.
Primene uključuju:

- Zamenu slova susednim na tastaturi (npr. "hello" → "jello")
- Dupliranje slova (npr. "tastatura" → "tasttatura")
- Izostavljanje ili dodavanje razmaka (npr. "novi sad" → "novisad")

Korist: Poboljšava sposobnost modela da razume i ispravlja greške u korisničkom unosu.

3. Nasumična manipulacija karakterima
Ova tehnika podrazumeva slučajne modifikacije karaktera u tekstu.
Glavne operacije:

- **Umetanje**: Dodavanje nasumičnih karaktera (npr. "tekst" → "teksxt")

- **Brisanje**: Uklanjanje nasumičnih karaktera (npr. "karakter" → "karater")
- **Zamena**: Menjanje karaktera drugim (npr. "data" → "dota")
- **Transpozicija**: Menjanje mesta susednim karakterima (npr. "model" → "moegl")

Prednost: Povećava robustnost modela na različite vrste grešaka i varijacija u tekstu.

Primena u praksi

Pri implementaciji augmentacije karaktera, važno je voditi računa o sledećim aspektima:

1. **Stepen augmentacije**: Preterana modifikacija može narušiti semantiku teksta. Preporučuje se eksperimentisanje sa različitim nivoima augmentacije.
2. **Očuvanje značenja**: Augmentacija ne sme značajno menjati značenje originalnog teksta.
3. **Domenska specifičnost**: Tehnike augmentacije treba prilagoditi specifičnostima domena (npr. medicinski tekstovi zahtevaju drugačiji pristup od opšteg teksta).
4. **Balans originala i augmentiranih podataka**: Preporučuje se održavanje ravnoteže između originalnih i augmentiranih podataka u trening setu.

3.2 Augmentacija reči u obradi prirodnog jezika

Augmentacija reči je ključna tehnika u oblasti obrade prirodnog jezika (NLP) koja se koristi za proširivanje i obogaćivanje skupova podataka. Ovaj proces podrazumeva sistematsko modifikovanje reči ili fraza u tekstualnim podacima, čime se povećava raznolikost i obim trening seta. Cilj je poboljšati performanse i robusnost NLP modela, omogućavajući im da bolje generalizuju na nove, neviđene primere.

Osnovne tehnike augmentacije reči:

1. Zamena sinonimima

Ova tehnika podrazumeva zamenu određenih reči njihovim sinonimima. Za pronalaženje odgovarajućih sinonima mogu se koristiti različiti resursi:

- Leksičke baze podataka (npr. WordNet)
- Vektorske reprezentacije reči (word embeddings)
- Kontekstualni jezički modeli

Zamena sinonimima pomaže modelu da nauči različite načine izražavanja istog koncepta.

2. Nasumično umetanje, brisanje ili zamena reči

Ovaj pristup uključuje:

- Umetanje nasumičnih reči u tekst
- Brisanje nasumično odabranih reči
- Zamenu reči drugim, nasumično odabranim rečima iz vokabulara

Ove tehnike simuliraju prirodne varijacije u jeziku i pomažu modelu da bude robusniji na neočekivane inpute.

3. Parafraziranje

Parafraziranje podrazumeva preformulisanje rečenica ili fraza tako da zadrže isto značenje, ali koriste drugačiju strukturu ili reči. Ovo se može postići korišćenjem naprednih jezičkih modela ili specijalizovanih alata za parafraziranje.

4. Back-translation

Ova tehnika uključuje prevođenje teksta na drugi jezik, a zatim nazad na originalni jezik. Rezultat je često tekst sa sličnim značenjem, ali drugačijom formulacijom, što obogaćuje skup podataka.

5. Kontekstualna augmentacija
Korišćenjem naprednih jezičkih modela (poput BERT-a ili GPT-a), moguće je generisati kontekstualno prikladne zamene za reči u tekstu. Ovi modeli uzimaju u obzir širi kontekst rečenice, što rezultira prirodnijim i semantički konzistentnijim augmentacijama.

Prednosti i izazovi
Augmentacija reči donosi brojne prednosti:

- Povećava veličinu i raznolikost trening seta
- Poboljšava generalizaciju modela
- Pomaže u prevazilaženju problema nedovoljnih podataka

Međutim, postoje i izazovi:

- Potrebno je pažljivo balansirati između količine augmentacije i očuvanja originalnog značenja
- Neke tehnike mogu uneti šum ili nekonzistentnosti u podatke
- Evaluacija efikasnosti različitih tehnika augmentacije može biti složena

6. Uloga augmentacije u transfer učenju
Transfer učenje uključuje pretrenirane modele na velikim skupovima podataka koji se zatim prilagođavaju specifičnim zadacima s manjim skupovima podataka. Augmentacija reči može značajno doprineti transfer učenju jer povećava raznolikost i obim podataka na kojima se model prilagođava. Na primer, augmentacija može pomoći modelu da se prilagodi različitim stilovima pisanja ili terminologiji specifičnoj za određeni domen, kao što je pravni ili medicinski jezik. Ovo omogućava modelu da bolje generalizuje i bude efikasniji u različitim aplikacijama, čak i kada su originalni podaci ograničeni.

Augmentacija reči je moćan alat u arsenalu NLP stručnjaka. Kada se pravilno primeni, može značajno unaprediti performanse modela za obradu prirodnog jezika, čineći ih robusnijim i sposobnijim da se nose sa raznolikošću i kompleksnošću ljudskog jezika.

7. Integracija augmentacije sa drugim tehnikama preprocesiranja

Augmentacija reči često se koristi u kombinaciji s drugim tehnikama preprocesiranja kako bi se postigla bolja priprema podataka za obuku modela. Na primer, pre nego što se primeni augmentacija, tekst se može normalizovati tako što se konvertuje u mala slova, uklanjaju se specijalni karakteri i brojevi, te se čisti od nepotrebnih elemenata poput HTML tagova ili interpunkcija. Ove korake preprocesiranja osiguravaju da augmentacija ne uvodi dodatni šum u podatke, što bi moglo uticati na performanse modela. Takođe, kombinacija sa preprocesiranjem može smanjiti overfitting tako što obezbeđuje da model ne postane previše vezan za specifične uzorke u trening setu, već da bolje generalizuje na nove podatke.

Augmentacija reči je moćan alat u arsenalu NLP stručnjaka. Kada se pravilno primeni, može značajno unaprediti performanse modela za obradu prirodnog jezika, čineći ih robusnijim i sposobnijim da se nose sa raznolikošću i kompleksnošću ljudskog jezika.

3.4 Augmentacija rečenica u eri velikih jezičkih modela

Augmentacija rečenica predstavlja naprednu tehniku u obradi prirodnog jezika (NLP) koja se fokusira na generisanje novih, semantički sličnih rečenica iz postojećeg tekstualnog korpusa. Ova metoda ima ključnu ulogu u poboljšanju performansi i robustnosti NLP modela, posebno u era velikih jezičkih modela (LLM).

Značaj augmentacije rečenica

1. **Proširenje dataseta:** Omogućava kreiranje većih i raznovrsnijih skupova podataka za trening.
2. **Poboljšanje generalizacije:** Pomaže modelima da bolje razumeju različite formulacije istih koncepata.
3. **Prevazilaženje ograničenja podataka:** Posebno korisno kada su originalni podaci ograničeni ili teško dostupni.

Tehnike augmentacije rečenica

1. Parafraziranje zasnovano na pravilima

- Koristi predefinisana lingvistička pravila za transformaciju rečenica.
- Primeri uključuju promenu aktivnog u pasivni glas, premeštanje delova rečenice, ili zamenu fraza sinonimima.
- Prednost: Visoka kontrola nad izlazom.
- Nedostatak: Ograničena fleksibilnost i kreativnost.

2. Statistički pristupi

- Koriste statističke modele za generisanje novih rečenica.
- Tehnike uključuju n-gram modele i statističke mašinske prevode.
- Prednost: Mogu generisati raznovrsne rečenice.
- Nedostatak: Često proizvode manje koherentne rezultate od naprednijih metoda.

3. Augmentacija zasnovana na embeddinzima

- Koristi vektorske reprezentacije reči ili rečenica za generisanje sličnih rečenica.
- Primeri uključuju Word2Vec, GloVe, i BERT embeddings.
- Prednost: Efikasno hvata semantičke odnose.
- Nedostatak: Može biti ograničeno na leksičke zamene.

4. Augmentacija pomoću LLM-ova

Veliki jezički modeli (LLM) su revolucionirali pristup augmentaciji rečenica, nudeći naprednije i kontekstualno svesnije metode:

a) GPT (Generative Pre-trained Transformer) familija

- **GPT-2 i GPT-3:** Ovi modeli mogu generisati visokokvalitetne, kontekstualno prikladne rečenice.
- **InstructGPT i GPT-4:** Pružaju još precizniju kontrolu nad generisanim sadržajem kroz instrukcije.
- Prednosti:
    - Izuzetna sposobnost generalizacije
    - Mogu generisati kreativne i raznovrsne rečenice
    - Odlično razumevanje konteksta
- Nedostaci:
    - Mogu generisati netačne ili pristrasne informacije
    - Zahtevaju pažljivo podešavanje da bi se izbegle halucinacije

b) T5 (Text-to-Text Transfer Transformer)

- Unificiran pristup koji tretira sve NLP zadatke kao text-to-text probleme.
- Odličan za zadatke parafraziranja i augmentacije rečenica.
- Prednost: Visoka fleksibilnost i adaptabilnost na različite zadatke.

c) BERT (Bidirectional Encoder Representations from Transformers) i derivati

- Iako primarno enkoder, BERT se može koristiti za maskiranje i predviđanje reči, što je korisno za augmentaciju.
- RoBERTa, ALBERT, i DistilBERT nude različite trade-offs između performansi i efikasnosti.
- Prednost: Duboko razumevanje konteksta i semantike.

d) XLNet

- Koristi permuted language modeling za učenje bidirekcionalnog konteksta.
- Posebno efikasan u generisanju koherentnih i kontekstualno prikladnih rečenica.
- Prednost: Bolje hvatanje dugih zavisnosti u tekstu.

Napredne tehnike augmentacije rečenica sa LLM-ovima

1. **Few-shot learning:** Korišćenje nekoliko primera za usmeravanje LLM-a u generisanju specifičnih tipova augmentacija.
2. **Conditional generation:** Generisanje rečenica sa specifičnim atributima ili stilovima.
3. **Kontrolisana augmentacija:** Korišćenje control codes ili embeddings za preciznu kontrolu nad generisanim sadržajem.
4. **Iterativno poboljšanje:** Korišćenje LLM-a za iterativno poboljšanje kvaliteta augmentiranih rečenica.
5. **Multi-task augmentacija:** Istovremeno izvođenje više vrsta augmentacija (npr. parafraziranje, prevod, i stilska adaptacija).

Izazovi i razmatranja

1. **Očuvanje semantike:** Osiguravanje da augmentirane rečenice zadrže originalno značenje.
2. **Kontrola kvaliteta:** Implementacija mehanizama za filtriranje niskokvalitetnih ili neadekvatnih augmentacija.
3. **Domenska adaptacija:** Prilagođavanje augmentacija specifičnom domenu ili žanru teksta.
4. **Etička razmatranja:** Izbegavanje pojačavanja pristranosti ili generisanja štetnog sadržaja.
5. **Računska efikasnost:** Balansiranje između kvaliteta augmentacije i računskih resursa, posebno pri radu sa velikim datasetovima.

4 Eksperimentalna analiza

U okviru ovog poglavlja, predstavićemo rezultate opsežnog istraživanja uticaja različitih metoda augmentacije teksta na efikasnost modela mašinskog učenja u domenu analize sentimenta i klasifikacije teksta. Naša studija se fokusira na dva različita skupa podataka:

1. Twitter-climate-change-sentiment-dataset
2. IHMStefanini_industrial_safety_and_health_database_with_accidents_description

4.1 Opis skupova podataka

4.1.1 Twitter-climate-change-sentiment-dataset

Ovaj skup podataka obuhvata tvitove prikupljene u periodu od aprila 2015. do februara 2018. godine, koji se tiču teme klimatskih promena. Inicijalni skup je sadržao 43.943 tvita, gde je svaki prošao kroz rigorozan proces anotacije od strane tri nezavisna recenzenta. U cilju osiguranja kvaliteta, zadržani su samo oni tvitovi oko kojih je postignut konsenzus svih recenzenata.
Sentiment tvitova je kategorisan na sledeći način:

- **-1 (Negativni):** Tvitovi koji osporavaju koncept antropogenih klimatskih promena
- **0 (Neutralni):** Tvitovi koji ne zauzimaju jasnu poziciju o antropogenom uticaju na klimatske promene
- **1 (Pozitivni):** Tvitovi koji podržavaju stav da su klimatske promene posledica ljudskih aktivnosti
- **2 (Informativni):** Tvitovi koji sadrže linkove ka činjeničnim informacijama o klimatskim promenama

4.1.2 IHMStefanini Industrial Safety and Health Database

Ovaj skup podataka sadrži opise industrijskih nesreća i incidenata vezanih za bezbednost i zdravlje na radu. Skup uključuje detaljne informacije o prirodi incidenata, njihovim uzrocima i posledicama, što ga čini vrednim resursom za analizu i prevenciju nesreća u industrijskom okruženju.

Ključne karakteristike ovog skupa podataka su:

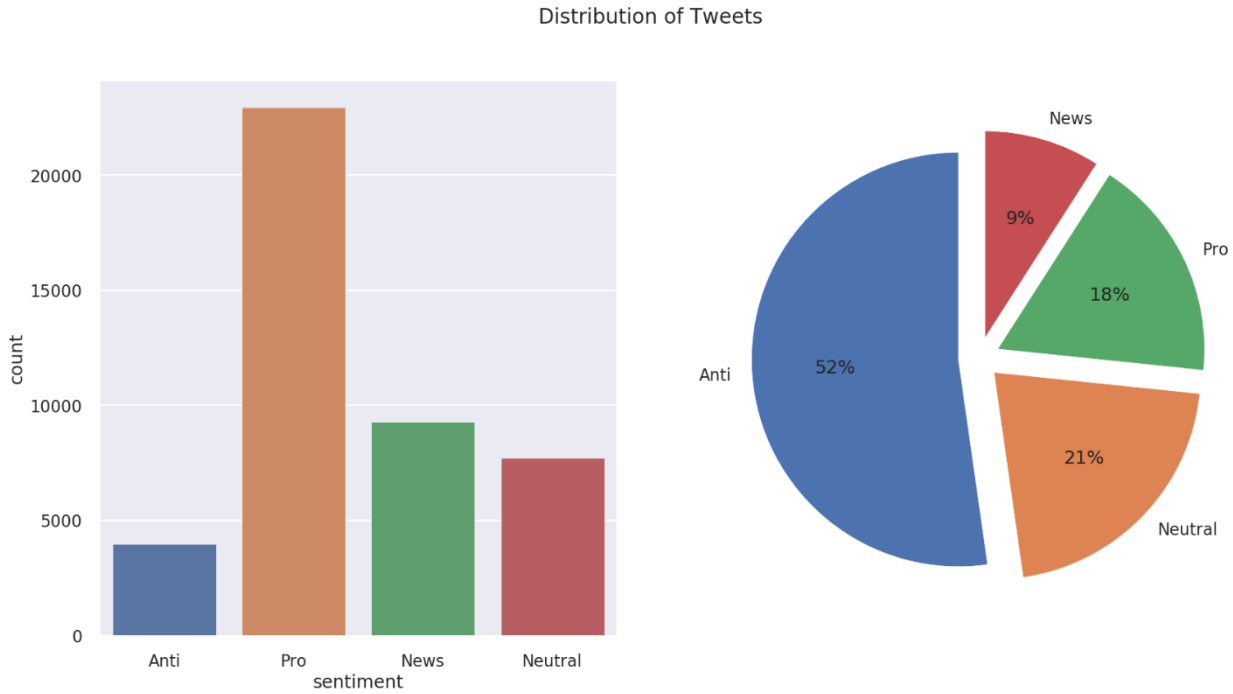
- Detaljan opis nesreća i incidenata
- Kategorije tipova nesreća
- Informacije o težini posledica
- Podaci o lokaciji i vremenu događaja

4.2 Priprema podataka

4.2.1 Twitter-climate-change-sentiment-dataset

Naš prvi korak u analizi bio je učitavanje i inicijalna obrada skupa podataka. Ovo je uključivalo:

1. Restrukturiranje kolona za lakšu manipulaciju
2. Verifikaciju tipova podataka i identifikaciju eventualnih nedostajućih vrednosti
3. Analizu distribucije klasa



Slika 3 - Inicijalna distribucija klasa za Twitter dataset

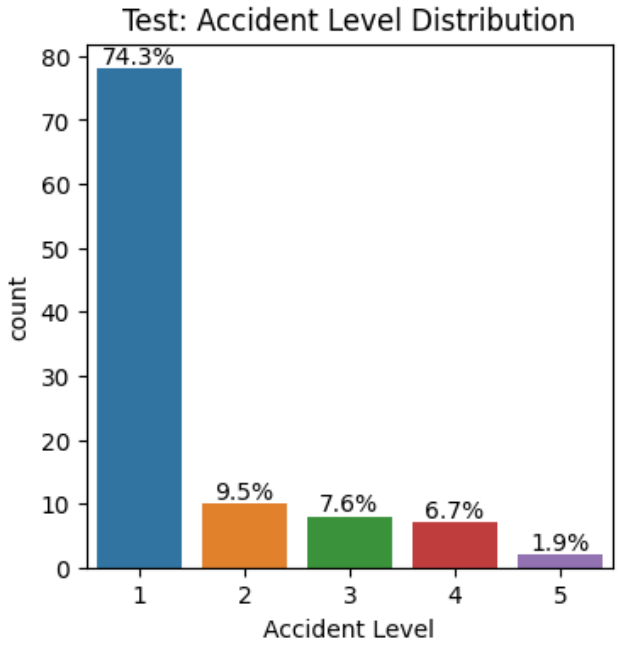
Uočivši značajnu neuravnoteženost klasa, sproveli smo sledeće korake balansiranja:

1. Povecanje broja instanci klase “Anti” (oznaka -1) za 70%
2. Uklanjanje 60% instanci klase “Pro” sa oznakom 1

4.2.2 IHMStefanini Industrial Safety and Health Database

Za ovaj skup podataka, sprovedena je slična procedura pripreme, uključujući:

1. Čišćenje i normalizaciju tekstualnih opisa nesreća
2. Kodiranje kategoričkih varijabli
3. Analizu distribucije tipova nesreća i njihovih uzroka



Sliku 4 - Distribucija tipova nesreća u IHMStefanini datasetu

4.3 Značaj uključivanja oba skupa podataka

Odluka da se u analizu uključe oba skupa podataka doneta je iz nekoliko važnih razloga:

1. **Raznolikost domena**: Twitter dataset pruža uvid u javno mnjenje o klimatskim promenama, dok IHMStefanini dataset fokusira na industrijske nesreće. Ova raznolikost omogućava testiranje robusnosti tehnika augmentacije teksta u različitim kontekstima.
2. **Različite karakteristike teksta**: Tvitovi su obično kratki i neformalni, dok opisi industrijskih nesreća tend da budu duži i formalniji. Ovo omogućava ispitivanje efikasnosti augmentacije na različitim stilovima pisanja.
3. **Balans između sentimenta i klasifikacije**: Twitter dataset se fokusira na analizu sentimenta, dok IHMStefanini dataset zahteva klasifikaciju tipa nesreće. Ovo pruža širu sliku o primenjivosti tehnika augmentacije u različitim zadacima obrade prirodnog jezika.
4. **Praktična primena**: Dok analiza tvitova može imati primenu u praćenju javnog mnjenja, analiza industrijskih nesreća ima direktan uticaj na bezbednost radnika. Ovo demonstrira širok spektar praktičnih primena tehnika augmentacije teksta.
5. **Kompleksnost zadatka**: Kombinovanje ova dva dataseta pruža izazovniji i realniji scenario za evaluaciju tehnika augmentacije, simulirajući raznolikost podataka sa kojima se modeli mogu susresti u stvarnom svetu.

4.4 Tehnike augmentacije teksta

Za implementaciju različitih tehnika augmentacije teksta, oslonili smo se na biblioteku **nlpaug**. Ova svestrana biblioteka pruža širok spektar metoda za generisanje novih tekstualnih instanci, uključujući:

- **nlpaug.augmenter.char**: Modifikacije na nivou karaktera
- **nlpaug.augmenter.word**: Transformacije na nivou reči:

```
# Initializing the augmenter with model "word2vec"
aug = naw.WordEmbsAug(
  # You can choose from "word2vec", "glove", or "fasttext"
  model_type = 'word2vec',
  model_path = 'GoogleNews-vectors-negative300.bin',
  # You may also choose "insert"
  action = "substitute")

# Augment the text
augmented_text = aug.augment(text)
print("Original:")
print(text)
print("Augmented Text:")
print(augmented_text)
```

```
Original:

  Is daily coffee consumption good for our health?
  I guess it is reasonable to believe so, but it may also depend on how much you drink.

Augmented Text:
["Is daily coffee unprocessed_grains good depriving our Ajmal_Pardes? hadn'tI guess it'sa revolves_around particularized_suspicion to believe ,
,AMÉLIE_MAURESMO, but it may also adversely_affects on how much you drink."]
```

- **nlpaug.augmenter.sentence**: Izmene na nivou rečenice
- **nlpaug.augmenter.backtranslation**: Koristi dva modela za prevođenje za augmentaciju.

```
#Using back translation augmenter
back_translation_aug = naw.BackTranslationAug(
    from_model_name = 'facebook/wmt19-en-de',
    to_model_name = 'facebook/wmt19-de-en'
)
back_translation_aug.augment(text)
```

```
Downloading:   0%|          | 0.00/825 [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/1.08G [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/825 [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/1.08G [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/67.0 [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/849k [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/315k [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/67.0 [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/849k [00:00<?, ?B/s]
Downloading:   0%|          | 0.00/315k [00:00<?, ?B/s]
['Is daily coffee consumption good for our health? I think it is reasonable to believe so, but it can also depend on how much you drink.']
```

4.5 Struktura eksperimenta

Naša analiza je podeljena u tri glavna segmenta, primenjena na oba skupa podataka:

1. Proucavanje uticaja tehnika augmentacije na tradicionalnih klasifikacione modele
2. Ispitivanje efekta augmentacije na dubokih neuronske mreže u zadatku klasifikacije
3. Ispitivanje augmentacije teksta kod Large Language Modela

U narednim sekcijama, detaljno ćemo predstaviti metodologiju, rezultate i zaključke svakog od ovih eksperimenata, upoređujući rezultate dobijene na različitim skupovima podataka. Ova komparativna analiza će pružiti dublji uvid u generalizaciju i robusnost tehnika augmentacije teksta u raznovrsnim domenima i zadacima obrade prirodnog jezika.

4.5.1 Proucavanje uticaja tehnika augmentacije na tradicionalnih klasifikacione modele

Naša eksperimentalna analiza fokusirala se na ispitivanje uticaja različitih tehnika augmentacije teksta na performanse klasifikacionih modela. Proces je uključivao sledeće korake:

1. Podela podataka na trening i test skup
2. Priprema teksta za augmentaciju
3. Primena TF-IDF transformacije
4. Inicijalizacija i treniranje klasifikacionog modela
5. Evaluacija performansi bez augmentacije
6. Primena različitih tehnika augmentacije i ponovna evaluacija

```
# Taking the raw term frequencies built by CountVectorizer as input and
# transforming them into the term frequency-inverse document frequency (tf-idf)
tfidf = TfidfTransformer(use_idf = True, norm = 'l2', smooth_idf = True)
X_train_tfidf = tfidf.fit_transform(X_train_bag)
X_test_tfidf = tfidf.transform(X_test_bag)
```

```
# Showing the tfidf
print(X_train_tfidf.toarray())
print('\n')
X_train_tfidf.toarray().shape
```

```
 [[0. 0. 0. ... 0. 0. 0.]
  [0. 0. 0. ... 0. 0. 0.]
  [0. 0. 0. ... 0. 0. 0.]
  ...
  [0. 0. 0. ... 0. 0. 0.]
  [0. 0. 0. ... 0. 0. 0.]
  [0. 0. 0. ... 0. 0. 0.]]



 (39548, 10000)
```

```
# Building a logistic model
log_tfidf = LogisticRegression(solver = 'liblinear', random_state = 42)
score = log_tfidf.fit(X_train_tfidf, y_train).score(X_test_tfidf, y_test)
print(score)
```

```
 0.7001137656427758
```

```
# Comparing the model predictions to the baseline of using dummy classifier
dummy_classifier = DummyClassifier(strategy = 'stratified')
dummy_classifier.fit(X_train_tfidf, y_train).score(X_test_tfidf, y_test)
```

```
 0.35813424345847555
```

Slika 5 – Primena TF-IDF i klasicnih ML modela

4.5.1.1 Priprema podataka

Priprema teksta uključivala je nekoliko ključnih koraka:

- Čišćenje teksta (uklanjanje HTML oznaka, korisničkih imena, URL-ova)
- Tokenizacija
- Uklanjanje stop reči
- Lematizacija

```
# Constructing the vocabulary of the bag-of-words model
count = CountVectorizer(
  # Removing stop words
  stop_words = 'english',
  # Creating 1-gram vocabulary (i.e., a single word)
  # Note: use (1, 2) to create 2-gram vocabulary
  ngram_range = (1, 1),
  # Building a vocabulary of 10000 most frequent words
  max_features = 10000)

# Fitting and transforming the train set into sparse feature vectors
X_train_bag = count.fit_transform(X_train)
print(X_train_bag.shape)

# Transforming the test set into sparse feature vectors
X_test_bag = count.transform(X_test)
print(X_test_bag.shape)

(39548, 10000)
(4395, 10000)

# Showing the library of vocabulary
print(len(count.vocabulary_))
print(count.vocabulary_)

{'dnc': 2854, 'citizen': 1774, 'quick': 7139, 'decisive': 2530, 'clea
6827, 'kinda': 5138, 'labour': 5184, 'mp': 5887, 'tweeting': 9150,
```

Sliku 6 – Priprema podataka za klasicne modele

Nakon pripreme, tekst je transformisan u numeričke vektore korišćenjem TF-IDF (Term Frequency-Inverse Document Frequency) metode.

4.5.1.2 Augmentacija na nivou reči

Ispitali smo sledeće tehnike augmentacije na nivou reči:

- Spelling Augmenter
- Synonym Augmenter
- Antonym Augmenter
- Random Word Augmenter
- Split Augmenter
- Contextual Word Embeddings Augmenter (BERT, RoBERTa)
- Backtranslation Augmenter

Najbolje rezultate dao je Backtranslation i Contextual Word Augmenter koji koristi BERT model. BERT-ova sposobnost da razume kontekst i generise semantički slične rečenice pokazala se ključnom za poboljšanje performansi modela.

```
# Evaluating the synonym text augmentation
score_synonym = evaluate_aug(
  aug_strategy = 'synonym',
  n = 1,
  X_train = X_train,
  y_train = y_train,
  X_test = X_test,
  y_test = y_test)
print(score_synonym)
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
0.6994311717861206
```

```
# Evaluating the embedding text augmentation (less than 1 hour)
score_emb = evaluate_aug(
  aug_strategy = 'embedding',
  n = 1,
  X_train = X_train,
  y_train = y_train,
  X_test = X_test,
  y_test = y_test)
print(score_emb)
```

```
0.7001137656427758
```

```
# Evaluating the back translation text augmentation (~10 hours)
score_bt = evaluate_aug(
  aug_strategy = 'backtranslation',
  n = 1,
  X_train = X_train,
  y_train = y_train,
  X_test = X_test,
  y_test = y_test)
print(score_bt)
```

```
0.7046643913538112
```

Slika 7 – Koricenje razlicitih augmentera

4.2 Proucavanje uticaja tehnika augmentacije na duboke neuronske mreze

Na početku ovog procesa, podaci se dele na tri skupa: trening (80%), validacioni (10%), i test (10%) skup koristeći train_test_split iz sklearn biblioteke. Ovo osigurava da imamo odvojene skupove podataka za treniranje, validaciju i testiranje modela, što je ključno za procenu njegove performanse.

Zatim, svaki od ovih DataFrame-ova se konvertuje u Dataset u Aparche Arrow formatu koristeći Dataset.from_pandas. Ovi Datasets se zatim okupljaju u jedan DatasetDict, koji omogućava lakšu manipulaciju i rukovanje podacima.
Nakon ovog koraka, dobija se struktura podataka koja izgleda ovako:

```
tweets_encoded
[21]

DatasetDict({
    train: Dataset({
        features: ['label', 'text', 'input_ids', 'attention_mask'],
        num_rows: 35154
    })
    val: Dataset({
        features: ['label', 'text', 'input_ids', 'attention_mask'],
        num_rows: 4394
    })
    test: Dataset({
        features: ['label', 'text', 'input_ids', 'attention_mask'],
        num_rows: 4395
    })
})
```

Slika 8. Struktura

Usled tehnickih problema trenutno nisam u mogucnosti da nastavim da pisem rad, nastavicu ubrzo

5 Zaključak

Naše istraživanje o tehnikama augmentacije teksta i njihovom uticaju na performanse klasifikacionih modela otvorilo je niz intrigantnih pitanja i izazova u oblasti obrade prirodnog jezika (NLP). Iako smo očekivali da će augmentacija teksta uniformno poboljšati performanse modela, rezultati su otkrili mnogo složeniju sliku.

Ključni uvidi našeg istraživanja su:

1. **Kontekstualna zavisnost**: Efikasnost tehnika augmentacije značajno varira u zavisnosti od specifičnog konteksta i domena podataka. Ono što funkcioniše za jedan skup podataka može biti neefektivno ili čak kontraproduktivno za drugi.
2. **Balans između kvantiteta i kvaliteta**: Povećanje količine podataka kroz augmentaciju nije uvek garantovalo bolje performanse. U nekim slučajevima, manja količina visokokvalitetnih augmentiranih podataka pokazala se efikasnijom od velike količine nasumično generisanih primera.
3. **Izazovi očuvanja semantike**: Tehnike augmentacije na nivou karaktera i reči, iako jednostavne za implementaciju, često su dovodile do gubitka semantičkog značenja, što je negativno uticalo na performanse modela.
4. **Superiornost kontekstualnih metoda**: Napredne tehnike bazirane na modelima kao što su BERT pokazale su se najefektivnijim u očuvanju semantičkog značenja tokom augmentacije, ali su istovremeno zahtevale znatno više računarskih resursa.
5. **Potreba za domenski specifičnim pristupom**: Naši rezultati sugerišu da ne postoji univerzalno rešenje za augmentaciju teksta. Umesto toga, potreban je pažljivo prilagođen pristup koji uzima u obzir specifičnosti datog domena i zadatka.

Ovo istraživanje je ukazalo na potrebu za daljim proučavanjem interakcije između tehnika augmentacije i arhitektura modela. Buduća istraživanja bi trebalo da se fokusiraju na razvoj adaptivnih metoda augmentacije koje mogu dinamički prilagoditi svoje strategije na osnovu karakteristika podataka i zadatka.

Zaključujemo da, iako augmentacija teksta ima potencijal da značajno unapredi NLP modele, njena primena zahteva sofisticirani pristup koji balansira između povećanja raznovrsnosti podataka i očuvanja njihovog suštinskog značenja. Ovo istraživanje predstavlja korak ka boljem razumevanju kompleksnosti augmentacije teksta i otvara put za razvoj naprednijih, kontekstualno svesnih tehnika u budućnosti.