MATNLI MA'LUMOTLARGA DASTLABKI ISHLOV BERISH BOSQICHLARI VA USULLARI

Fazliddinov Xolmurod Dilmurod o'g'li
FarDU Axborot tizimlari va texnologiyalari yo'nalishi 3-kurs talabasi
xolmurodfazliddinov@gmail.com
Abdukadirov Baxtiyor Abduvaxitovich
Fardu Axborot Texnologiyalari Kafedrasi Dotsenti
bakxtiyor.uz@gmail.com

Anotatsiya: *Mazkur ishda matnli ma'lumotlarga dastlabki ishlov berish bosqichlari va usullari yoritilgan. Xususan, tokenizatsiya, stemming, lemmatizatsiya, stop-so'zlarni olib tashlash hamda matnni normalizatsiya qilish jarayonlari ko'rib chiqilgan. Shuningdek, matnli ma'lumotlarni raqamli ko'rinishga o'tkazish uchun vektorlashtirish usullari — Bag-of-Words, TF-IDF va zamonaviy Word Embeddings modellari tahlil qilingan. Python dasturlash tilining NLTK, SpaCy va Scikit-learn kabi kutubxonalari yordamida ushbu jarayonlarni amalga oshirish imkoniyatlari ko'rsatib berilgan. Ishda matnli ma'lumotlarni to'g'ri qayta ishlashning tahlil natijalarining aniqligi va ishonchliligiga ta'siri asoslab berilgan.*

Kalit so'zlar: *matnli ma'lumotlar, NLP, tokenizatsiya, stemming, lemmatizatsiya, stop-so'zlar, vektorlashtirish, Bag-of-Words, TF-IDF, Word2Vec, SpaCy, NLTK, Python*

STAGES AND METHODS OF TEXT DATA PREPROCESSING

Fazliddinov Xolmurod Dilmurod Ogli
Fergana State University, Information Systems and Technologies, 3rd-Year Student
Email: xolmurodfazliddinov@gmail.com
Abdukadirov Bakhtiyor Abduvakhitovich
Fergana State University, Associate Professor Of The Department Of Information Technologies
Email: bakxtiyor.uz@gmail.com

Annotation: *This paper discusses the stages and methods of preprocessing textual data. In particular, processes such as tokenization, stemming, lemmatization, stop-word removal, and text normalization are examined. Additionally, vectorization methods for converting text data into numerical form—such as Bag-of-Words, TF-IDF, and modern Word Embeddings models—are analyzed. The possibilities of implementing these processes using Python libraries such as NLTK, SpaCy, and Scikit-learn are demonstrated.*

The study also justifies the impact of proper text preprocessing on the accuracy and reliability of analytical results.

Keywords: *text data, NLP, tokenization, stemming, lemmatization, stop words, vectorization, Bag-of-Words, TF-IDF, Word2Vec, SpaCy, NLTK, Python*

ЭТАПЫ И МЕТОДЫ ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ТЕКСТОВЫХ ДАННЫХ

Фазлиддинов Холмурод Дилмурод Угли
Фергу, Направление «Информационные Системы И Технологии», Студент 3 Курса
Email: xolmurodfazliddinov@gmail.com
Абдукадиров Бахтиёр Абдувахитович
Фергу, Доцент Кафедры Информационных Технологий
Email: bakxtiyor.uz@gmail.com

Аннотация: *В данной работе рассмотрены этапы и методы предварительной обработки текстовых данных. В частности, изучены процессы токенизации, стемминга, лемматизации, удаления стоп-слов и нормализации текста. Также проанализированы методы векторизации текстовых данных, такие как Bag-of-Words, TF-IDF и современные модели Word Embeddings. Показаны возможности реализации данных процессов с использованием библиотек Python, таких как NLTK, SpaCy и Scikit-learn. Обосновано влияние корректной обработки текстовых данных на точность и надежность результатов анализа.*

Ключевые слова: *текстовые данные, NLP, токенизация, стемминг, лемматизация, стоп-слова, векторизация, Bag-of-Words, TF-IDF, Word2Vec, SpaCy, NLTK, Python*

Kirish: Bugungi raqamli asrda matnli ma'lumotlarning hajmi misli ko'rilmagan darajada o'sib bormoqda. Internet, ijtimoiy tarmoqlar, elektron pochta va turli hujjatlar orqali kuniga petabaytlab ma'lumot hosil bo'lmoqda. Ushbu ulkan ma'lumotlar oqimini samarali tahlil qilish, undan qimmatli bilimlarni ajratib olish va qaror qabul qilish jarayonlarida qo'llash zamonaviy texnologiyalarning asosiy vazifalaridan biridir. Matnli ma'lumotlarga dastlabki ishlov berish bu jarayonning fundamental bosqichi hisoblanadi. Bu bosqich ma'lumotlarning tozaligini, izchilligini va kelgusi tahlil uchun tayyorligini ta'minlaydi. Ushbu ishning maqsadi Python dasturlash tilida matnli ma'lumotlarga dastlabki ishlov berishning asosiy bosqichlari va usullarini chuqur o'rganish, ularni amaliy misollar yordamida ko'rsatib berishdan iborat. Python o'zining boy kutubxonalari, xususan, NLTK, SpaCy va Gensim kabi vositalari bilan matn tahlili sohasida keng qo'llaniladi. Mazkur

ishda matnni tokenizatsiya qilish, stemming va lemmatizatsiya, stop-so'zlarni olib tashlash, ma'lumotlarni normalizatsiya qilish va boshqa muhim jarayonlar batafsil yoritiladi. Mavzuning dolzarbligi shundaki, sifatsiz yoki noto'g'ri ishlov berilgan matnli ma'lumotlar asosida qilingan tahlillar noto'g'ri xulosalarga olib kelishi mumkin. Shu sababli, ma'lumotlarni dastlabki qayta ishlash tahlilning aniqligi va ishonchliligini oshirishda hal qiluvchi rol o'ynaydi. Bu jarayon matn tasnifi, sentyment tahlili, ma'lumotlarni qidirish va boshqa tabiiy tilni qayta ishlash (NLP) vazifalarining samaradorligini sezilarli darajada yaxshilaydi.

Pythonda matnlar ustida amalga oshirish mumkin bo'lgan tayyor funksiyalar mavjud. Odatda, biror funksiya ma'lum bir obyektga (o'zgaruvchi, ma'lumot turiga) xos bo'lsa, bunday funksiyalar metodlar deb ataladi. Metodlarni qo'llash uchun metod nomi matndan so'ng .metod_nomi() ko'rinishida yoziladi. Keling, shunday metodlarning ba'zilari bilan tanishamiz.

Python nuqtayi nazaridan matn (string) shunchaki belgilar ketma-ketligidir. Tokenizatsiya — bu tartibsiz belgilar yig‘indisini kompyuter tushunadigan ob’ektlar ro‘yxatiga aylantirish bosqichlari 2 xildir bular:

1. **Mantiq:** Python dasturi matnni tahlil qilishdan oldin uni "bo‘laklarga" (atomlarga) bo‘lib chiqishi shart. Bu bo‘laklar asosida keyinchalik statistik tahlil yoki chastotali lug‘atlar tuziladi.
2. **Muammo:** Shunchaki bo'shliq bo'yicha ajratish (whitespace tokenization) yetarli emas. Masalan, "o'qidi." so'zidagi nuqtani so'zdan ajratish kerak, aks holda dastur "o'qidi" va "o'qidi." ni ikki xil so'z deb hisoblaydi.

Bu jarayon tabiiy tilni qayta ishlash (NLP) sohasi doirasida amalga oshiriladi, bu esa kompyuterlarga inson tilini tushunish, izohlash va hatto yaratish imkonini beradi. NLPning asosiy maqsadi — kompyuterlarning matnli ma'lumotlardan inson kabi ma'no chiqarish qobiliyatini oshirishdir. Masalan, sentiment tahlili orqali kompaniya mijozlarning mahsulotlari haqidagi fikrlarini aniqlay oladi, mavzu modellashtirish yordamida esa katta hajmdagi hujjatlar to'plamidan asosiy mavzularni ajratib olish mumkin. Matnli ma'lumotlarning ahamiyati bir qancha jihatlar bilan belgilanadi. Birinchidan, ular inson bilimi va tajribasining eng boy manbai hisoblanadi. Kitoblar, maqolalar va arxivlar asrlar davomida to'plangan axborotni o'z ichiga oladi. Ikkinchidan, zamonaviy biznesda matnli ma'lumotlar mijozlar bilan muloqot, bozor tahlili va raqobat ustunligini ta'minlashda hal qiluvchi rol o'ynaydi. Kompaniyalar ijtimoiy tarmoqlardagi sharhlarni tahlil qilib, mijozlarning ehtiyojlarini aniqlaydi va mahsulotlarini yaxshilaydi. Uchinchidan, ilmiy tadqiqotlarda matnli ma'lumotlar yangi kashfiyotlar qilish, nazariyalarni tasdiqlash yoki inkor etish uchun asos bo'lib xizmat qiladi. Tibbiyotda bemorlarning tibbiy kartalaridagi matnli ma'lumotlarni tahlil qilish orqali kasalliklarni aniqlash va davolash usullarini takomillashtirish mumkin. Matnli ma'lumotlarni tushunish nafaqat ularning lug'aviy

ma'nosini anglashni, balki kontekstual, semantik va pragmatik jihatlarni ham o'z ichiga oladi. Bir so'zning ma'nosi turli kontekstlarda o'zgarishi mumkin. Masalan, "bank" so'zi moliyaviy muassasani ham, daryo qirg'og'ini ham anglatishi mumkin. Kompyuter dasturlari ushbu nozikliklarni tushunishga o'rganishi kerak.

Tekstli ma'lumotlarga dastlabki ishlov berishda vektorlashtirish va xususiyatlarni ajratish muhim bosqichlardir. Bu jarayonlar tabiiy tilni qayta ishlashning (NLP) asosini tashkil etib, matnni mashinaga tushunarli raqamli formatga o'tkazish imkonini beradi. Vektorlashtirish matndagi so'zlar, jumlalar yoki butun hujjatlarni ko'p o'lchovli raqamli vektorlarga aylantirishni anglatadi. Bu raqamli ifodalar matnning semantik va sintaktik xususiyatlarini aks ettiradi. Eng oddiy vektorlashtirish usullaridan biri "Bag-of-Words" (So'zlar sumkasi) modelidir. Bu modelda har bir hujjat so'zlar chastotasi bo'yicha vektorga aylantiriladi.

Masalan: "Python dasturlash tili" va "Python juda mashhur til" jumlalarini ko'rib chiqaylik. Lug'at "Python", "dasturlash", "tili", "juda", "mashhur" so'zlaridan iborat bo'lsa, birinchi jumla [1, 1, 1, 0, 0] vektoriga, ikkinchi jumla esa [1, 0, 1, 1, 1] vektoriga aylantiriladi. Bu usul so'zlarning tartibini e'tiborga olmaydi, ammo uning soddaligi va samaradorligi tufayli keng qo'llaniladi. So'zlar sumkasi modelining kamchiliklarini bartaraf etish uchun "TF-IDF" (Term Frequency-Inverse Document Frequency) usuli joriy etilgan. TF-IDF nafaqat so'zning hujjatdagi chastotasini (TF), balki uning corpusdagi umumiy kamyobligini ham (IDF) hisobga oladi. Ya'ni, kamdan-kam uchraydigan, ammo ma'lum bir hujjatda tez-tez ishlatiladigan so'zlarga yuqori og'irlik beriladi, umumiy so'zlarga (masalan, "va", "bilan") esa past og'irlik beriladi. Bu esa har bir so'zning matnga qo'shgan informativ qiymatini aniqroq aks ettiradi. Python tilida Scikit-learn kutubxonasining TfidfVectorizer klassi bu jarayonni osonlashtiradi. Zamonaviy vektorlashtirish usullari orasida "Word Embeddings" (So'z o'rnatmalari) alohida o'rin tutadi. Word2Vec, GloVe va FastText kabi modellar so'zlarni zich vektorlarga o'tkazadi, bu vektorlar so'zlar orasidagi semantik munosabatlarni aks ettiradi. Masalan, "qirol" so'zining vektori "qirolicha" so'zining vektoridan "erkak" va "ayol" so'zlari orasidagi farqqa o'xshash munosabatda bo'ladi. Bu modellar katta miqdordagi matn korpuslari yordamida o'qitiladi va so'zlarning kontekstini tushunishga imkon beradi. Word Embeddings yordamida "qirol - erkak + ayol = qirolicha" kabi matematik operatsiyalarni bajarish mumkin, bu esa matnni tushunishda yangi imkoniyatlar ochadi. Xususiyatlarni ajratish esa matndan muhim va axborotga boy xususiyatlarni aniqlash va olish jarayonidir. Bu jarayon modelning samaradorligini oshirish va ortiqcha ma'lumotlarni kamaytirishga yordam beradi. Matnli ma'lumotlarda xususiyat sifatida so'zlar chastotasi, n-grammalar (birgalikda kelgan so'zlar ketma-ketligi, masalan, "Python dasturlash"), qism-nutq belgilarining (ot, fe'l, sifat) mavjudligi, sintezlangan so'zlar (masalan, stemming yoki lemmatizatsiyadan keyingi so'zlar) yoki hatto matnning uzunligi va so'zlar soni kabi

metama'lumotlar ishlatilishi mumkin. Masalan, spam filtrlashda elektron pochta xabaridagi bosh harflarning ko'pligi yoki ma'lum bir kalit so'zlarning mavjudligi xususiyat sifatida ajratilishi mumkin. Pythonning NLTK (Natural Language Toolkit) va SpaCy kutubxonalari matn xususiyatlarini ajratish uchun keng imkoniyatlar yaratadi. NLTK tokenizatsiya, stemming, lemmatizatsiya, qism-nutq belgilarini aniqlash kabi funksiyalarni taqdim etadi. SpaCy esa tezkor va aniq tokenizatsiya, nomlangan ob'ektni aniqlash (Named Entity Recognition - NER) va sintaktik tahlil kabi ilg'or imkoniyatlar bilan ajralib turadi.

Xulosa: Matnli ma’lumotlarga dastlabki ishlov berish zamonaviy axborot texnologiyalarida muhim va ajralmas bosqich hisoblanadi. Ushbu jarayon matnni tahlil qilishdan oldingi tayyorgarlik bosqichi bo‘lib, ma’lumotlarning sifatini oshirish, ortiqcha va keraksiz elementlardan tozalash hamda ularni bir xil standartga keltirishni ta’minlaydi. Ayniqsa, katta hajmdagi matnli ma’lumotlar bilan ishlashda dastlabki ishlov berishsiz aniq va ishonchli natijalarga erishish deyarli imkonsizdir.

Ish jarayonida ko‘rib chiqilgan tokenizatsiya usuli matnni kichik birliklarga ajratish orqali tahlilni soddalashtiradi. Stemming va lemmatizatsiya esa so‘zlarning turli shakllarini yagona asosga keltirib, ma’lumotlarning hajmini kamaytiradi va model samaradorligini oshiradi. Stop-so‘zlarni olib tashlash orqali esa matndan informativ qiymati past bo‘lgan elementlar chiqarib tashlanadi, bu esa hisoblash resurslaridan samarali foydalanishga yordam beradi. Normalizatsiya jarayoni esa matnni yagona ko‘rinishga keltirib, tahlil aniqligini oshiradi.

FOYDALANILGAN ADABIYOTLAR

1. Jurafsky D., Martin J.H. – *Speech and Language Processing* (3rd edition draft), 2023.
2. Bird S., Klein E., Loper E. – *Natural Language Processing with Python*, O’Reilly Media, 2009.
3. Manning C.D., Raghavan P., Schütze H. – *Introduction to Information Retrieval*, Cambridge University Press, 2008.
4. Mikolov T. et al. – *Efficient Estimation of Word Representations in Vector Space*, 2013.
5. Pedregosa F. et al. – *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research, 2011.
6. Honnibal M., Montani I. – *spaCy 2: Natural Language Understanding with Bloom Embeddings*, 2017.