ИЛМ-ФАН
ВА ИННОВАЦИОН РИВОЖЛАНИШ

НАУКА
И ИННОВАЦИОННОЕ РАЗВИТИЕ

SCIENCE
AND INNOVATIVE DEVELOPMENT

4 / 2022

ТОШКЕНТ – 2022

МУНДАРИЖА / СОДЕРЖАНИЕ / CONTENTS

02.00.00 КИМЁ ФАНЛАРИ ХИМИЧЕСКИЕ НАУКИ CHEMISTRY SCIENCES

05.00.00 ТЕХНИКА ФАНЛАРИ ТЕХНИЧЕСКИЕ НАУКИ TECHNICAL SCIENCES

Ключевые слова: *обработка естественного языка, NLP, spaCy, Python, части речи, лемматизация, токенизация, синтаксический анализатор, конвейерная архитектура.*

MAKING USE OF A 'SPACY' MODULE IN THE NATURAL LANGUAGE PROCESSING

Elov Botir Boltaevich,
Doctor of Philosophy in Technical Sciences (PhD), Associate Professor, Head of the Department of Computational Linguistics and Digital Technologies, Tashkent State University of Uzbek Language and Literature named after Alisher Navoi

Abstract. *This article discusses the use and tools of the spaCy module, which is written in Python machine language, in the Natural Language Processing (NLP), considered as one of the main areas of computer linguistics. A text in a natural language contains separate units (symbols) and can be divided into several interrelated parts belonging to different levels. The article, therefore, presents methods for tokenizing text using the spaCy library tools as well as the lemma, POS, tag, dep, shape, alpha, and stop attributes generated in a pipeline process.*

Keywords: *Natural language processing, NLP, spaCy, Python, part-of-speech, lemmatization, token, parser, pipeline architecture.*

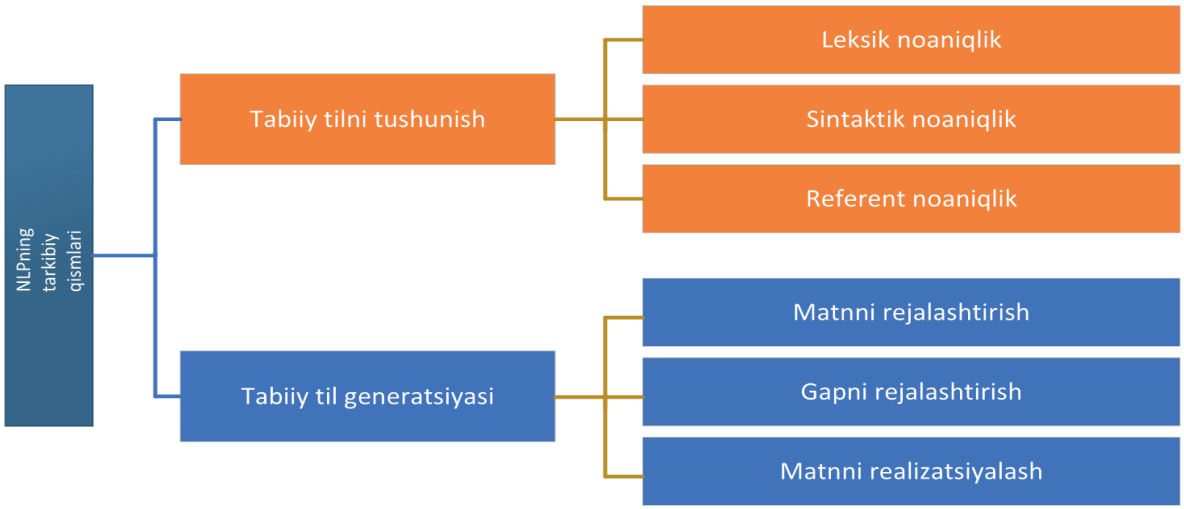
lash borasida dunyo miqyosida tez sur'atlarda yaratilayotgan til korpuslarining roli beqiyos.

Til korpuslari – til bo'yicha tadqiqot va amaliy topshiriqlar yechimi uchun zarur ish quroli. U oddiy elektron kutubxonadan farq qiladi. Elektron kutubxonaning maqsadi – xalqning ijtimoiy-siyosiy, ma'naviy, iqtisodiy hayotini aks ettiruvchi badiiy va publitsistik asarlarni nisbatan to'liq qamrab olish. Elektron kutubxona matnlari til nuqtayi nazaridan ishlov berilmaganligi sababli tadqiqotlar uchun noqulaylik tug'diradi. Chunki elektron kutubxona ilmiy tadqiqot materiali bazasini tayyorlash maqsadida tuzilmaydi, balki milliy ma'naviy merosni to'plashni maqsad qilgan bo'ladi. Til korpusi esa elektron kutubxonadan farqli o'laroq, tilni o'rganish va tadqiq qilish uchun foydali va qiziqarli matnlarni to'plashni nazarda tutadi. Til korpuslarini yaratish uchun NLP (tabiiy tilni qayta ishlash) vositalaridan foydalaniladi.

Tabiiy tilni qayta ishlash – bu nutq yoki matn orqali tabiiy tilni o'rganish va uni dasturiy ta'minot hamda algoritmlar yordamida qanday boshqarish yoki tushunish jarayoni hisoblanadi. NLPni o'rganish uzoq vaqtdan (50 yildan ortiq) beri mavjud bo'lib, matnni qayta ishlash va tushunish uchun qanday modellar yaratish mumkinligi bilan qiziqish uyg'otdi. Tabiiy tilni qayta ishlash uchun statistik ma'lumotlar va klassik tilshunoslik modellaridan foydalanishga e'tibor qaratilgan bo'lsa-da, hozirda NLPga asoslangan tizimlar uchun sun'iy intellekt texnologiyasi hisoblanmish neyro-tarmoqlardan foydalanilmoqda. Bugungi kunda ko'plab yirik tadqiqot markazlari allaqachon milliardlab so'zlarga (masalan, GPT-3 [1]) asoslangan til modellarini yaratishga e'tibor qaratgan.

Kompyuter lingvistikasida tabiiy tilni modellashtirish inson faoliyati davomida axborot almashinuvida foydaniladigan va ushbu faoliyatga asosan doimiy o'zgarib boradigan ko'p bosqichli belgilar tizimi bilan bog'liq jarayondir. Ma'lumotlarni tahlil qilishda an'anaviy vositalar, metodologiyalar va ko'nikmalardan foydalanish uchun berilgan matnni strukturlangan formatga o'tkazish lozim. Tabiiy tilni qayta ishlash (Natural Language Processing, NLP) platformasi moslashuvchan, mustahkam va samarali komponentlar to'plamini taqdim etadi [2, 76–89-b., 3, 143–150-b.]. Platforma asosida matnning har bir qismini turli usullar bilan boyitish uchun foydalaniladigan NLP mavjud bo'ladi. *NLP* – sun'iy intellektning kichik sohasi bo'lib, kompyuterlar va inson tillari o'rtasidagi o'zaro aloqani ta'minlaydi. NLP – bu kompyuterlar uchun inson tillarini *tahlil qilish, tushunish* va *ma'no chiqarish* jarayoni bo'lib, uni ikkita asosiy komponentga ajratish mumkin(1-rasm):

- *tabiiy tilni tushunish (Natural Language Understanding, NLU);*
- *tabiiy tilni generatsiya qilish (Natural Language Generation, NLG).*



1-rasm. NLP komponentlari

Ushbu maqolada NLP vazifalarini bajarishda mavjud qadamlar ko‘rib chiqiladi. Tabiiy tilning manbayi *nutq (tovush)* yoki *matn* bo‘lishi mumkin (2-rasm). Hozirda o‘zbek tilidagi matnlar va tovushni qayta ishlash tizimlari hamda dasturiy ta’minotlar ishlab chiqilmagan. O‘zbek tilidagi matnlarni qayta ishlash uchun ushbu maqolada keltirilgan usullar yordamida til modellarini yaratish, matnlarni *morfologik, leksik, sintaktik, pragmatik va semantik* tahlil qilishda Python vositalaridan foydalanish muhim ahamiyatga ega.

Dunyo tilshunosligida jadal rivojlanayotgan kompyuter lingvistikasi yo‘nalishida tabiiy tilga ishlov berish, so‘rov berish va javob olish sifatini yaxshilash, lingvistik tadqiqot, lingvodidaktika, leksikografiyada korpuslardan foydalanish dolzarb masalaga aylandi. Tilshunoslikda, xususan, kompyuter lingvistikasi sohasida korpus birliklarini tanlash, ularni lingvistik, jumladan, morfologik, semantik, sintaktik va stilistik teglash, stilistik teglar tizimini ishlab chiqish, til korpusi uchun maxsus axborotlarni qamrab olgan uslubiy xoslanishni farqlovchi lingvistik ta’minot yaratish muammosi dolzarb bo‘lib qolmoqda.

Til birliklarining xususiyatlarini aniqlash maqsadida elektron qidiruv imkoniyati mavjud, tabiiy tilning raqamlashgan yozma va og‘zaki matnlar jamlanmasidan iborat axborot tizimlarini ishlab chiqish bugungi kunda kompyuter lingvistikasi sohasida dolzarb muammo hisoblanadi.

Milliy korpus – til birligining o‘zgarishi, eskirishi, yangilarining paydo bo‘lishi, ma’nosining kengayishi va torayishi, yangi iboralarning paydo bo‘lishini kuzatish, an’anaviy va zamonaviy lug‘atlar tuzishda keng imkoniyatli dasturlashtirilgan tizim. Milliy korpusning boshqa tur til korpuslaridan farqi ta’limiy korpus, mualliflik korpusi, poetik matnlar korpusi, ilmiy va rasmiy matnlar korpusi, og‘zaki matnlar, badiiy matnlar va gazetalar hamda dialektlar korpusi kabi ichki korpuslarni o‘z tarkibiga qamrab olganligi bilan belgilanadi. Undan lingvistlar, leksikograflar, kompyuter lingvistlari, dasturchilar, muharrirlar, tarjimonlar, jurnalistlar, noshirlar, olimlar, o‘qituvchilar, ta’lim oluvchilar va boshqa har qanday soha mutaxassisi keng foydalanish imkoniyatiga ega. Korpussiz na nazariy, na amaliy filologiya taraqqiy etadi. Foydalanishda deyarli kasbiy tabaqalanishga yo‘l qo‘ymaydigan til korpuslari barcha fan-soha vakillarini birday qiziqtirishi tabiiy. Shuning uchun ham ta’lim korpusi, sheva matnlari korpusi, mualliflik korpusi, poetik matnlar korpusi, og‘zaki, ilmiy, rasmiy matnlar korpusi, gazeta matnlari korpusi kabi qator mikrokorpuslarning tuzilayotganligi ahamiyatlidir.

An’anaviy yondashuvlar semantikani o‘rnatishda gaplardan so‘zlarni ajratish va so‘z turkumlarini aniqlash uchun leksik va

sintaksis tahlilni o‘z ichiga oladi. C. Chantrapornchai va A. Tunsakul tomonidan matnlarni tahlil qilish maqsadida ishlatiladigan ikkita mashinali o‘rganishga asoslangan metodologiya taqdim etilgan [4, 108–120-b.] bo‘lib, metodologiyalar quyidagi vazifalarga asoslangan: *NERlarni aniqlash* va *matnni tasniflash*. Ular tomonidan taqdim etilgan metodologiya yordamida bir nechta qadamlar orqali gaplarda tokenlash va NER obyekti ajratib olingan. Keyingi qadamda esa obyekt turini tanib olish modeli ishlab chiqilgan bo‘lib, ushbu vazifalarning bajarilishini solishtirish uchun ikkita vosita – SpaCy va BERT ishlatilgan.

R. Yanti, I. Santoso va H. Suadaa tomonidan Indoneziyadagi provinsiyada sodir bo‘lgan elektr ta’minotidagi uzilishlar bilan bog‘liq Twitterda yozilgan ko‘plab shikoyatlarni avtomatik tahlil qiluvchi NLP dasturiy ta’minoti SpaCy paketi vositalari orqali ishlab chiqilgan [5, 76–86-b.]. M. Kharis va K. Laksono olib borgan tadqiqotlarga ko‘ra, nemis tilini o‘rganish bo‘yicha A1 darajasidagi CEFR standartlari darsliklarda mavjud bo‘lgan lug‘at turlari va tokenlar sonini solishtirish orqali hamda darsliklardagi “lemma”lar German Lemmatizerdan foydalanib, SpaCy paketi yordamida aniqlangan [6, 148–155-b.]. Tahlil natijalariga ko‘ra, SpaCy lemmatizatori so‘zlarning shaklini lemmatizatsiya va tahlil qilish hamda nemis tilidagi so‘z sinflarini tasniflash imkoniyatini taqdim etgan.

Katta hajmdagi (big data) matnning barcha muhim nuqtalarini saqlash va qayta ishlash orqali matn xulosasi deb ataladigan o‘zgartirilgan (rezyume) shaklini aniqlash ustida A. Kumar va V. Katiyar ilmiy izlanishlar olib borgan [7]. Ushbu vazifa juda murakkab bo‘lib, hujjatning katta hajmdagi qat’iy tahlilini talab qiladi. A. Kumar va V. Katiyar tomonidan taklif etilayotgan yondashuv asosida tabiiy tilni qayta ishlashning ikkita: STANZA va SPACY paketlaridan foydalanilgan. Har ikkala paketlar ham zamonaviy texnologiyalarga asoslangan va HINDI tilini qayta ishlash uchun mos vositalarga ega bo‘lib, olingan natijalar o‘zaro qiyosiy taqqoslangan. Yuqoridagi ilmiy izlanishlarni tahlil qilish natijasida tabiiy tildagi matnni alohida birlik (belgi)lardan iborat lingvistik sathlarga mutanosib ravishda turli bosqichlarga mansub o‘zaro bog‘liq bir qancha qismlarga ajratish mumkin [2, 76–89-b.; 3, 143–150-b.; 7, 2023–2030-b.; 8, 1–5-b.; 9, 72–79-b.; 10, 119–122-b.].

Material va metodlar

Maqolada matnni qayta ishlash uchun zarur qadamlar ketma-ketligi Python tilidagi Spacy paketi vositalari orqali misollar bilan keltirilgan. Maqolada keltirilgan usullar milliy korpusni shakllantirishda muhim vosita bo‘lib xizmat qiladi.

Bugungi kunda NLP masalalarini hal qilishda jahonda ko‘plab kutubxonalar, paketlar, texnologiyalar mavjud. Ularning har biri o‘zining ijobiy va salbiy tomonlariga ega. Python NLP masalalarini hal qilishda eng mos keluvchi kutubxonalarga ega tildir. Har bir NLP kutubxonalari ma’lum maqsadlar bilan qurilgan, shu bois bitta kutubxona hamma muammolar uchun yechimlar taqdim etmasligi aniq [11; 12 2–6-b.; 13, 411–420-b.]. Dasturchi ushbu kutubxonalardan qachon va qayerda, nimadan foydalanishini aniq bilishi lozim. Ushbu maqolada spaCy kutubxonasi vositalari orqali NLP masalalarini hal qilish usullarini ko‘rib chiqamiz [14, 324–337-b.; 15, 25–36-b.].

SpaCy Metyu Honnibal [13, 16] tomonidan ishlab chiqilgan va Ines Montani [13, 17] tomonidan qo‘shimcha modullari ishlab chiqilgan. Spacy kutubxonasida ingliz, hind, ispan, nemis, fransuz, golland tillari kabi 60 dan ortiq tillardagi matnlarni qayta ishlash mumkin. NLTK kutubxonasi, asosan, ilmiy tadqiqotlar va tabiiy tilni qayta ishlashni o‘rganish uchun ishlatiladi.

Tadqiqot natijalari

NLPda matnni qayta ishlash bosqichlari

NLP strukturlanmagan matndan zarur tushunchalarni olishga yordam beradi va turli amallarni bajarishga xizmat qiladi:

– *avtomatik umumlashtirish;*
– *nomlangan obyektni aniqlash;*
– *savol-javob tizimlari;*
– *hissiyotni tahlil qilish.*

ISSN 2181-9637
4 / 2022

2-rasm. NLP bosqichlari

SpaCy – bu Pythonda NLP uchun ochiq kodli kutubxona bo‘lib, ma’lumot olish yoki tabiiy tilni tushunish tizimlarini yaratishga mo‘ljallangan hamda qulay API xizmatlarini taqdim etadi. SpaCy matnlarni tahlil qilishda quyida keltirilgan imkoniyatlarni taqdim etadi:

– *buzilmaydigan tokenizatsiya;*
– *nomlangan obyektni tanib olish;*
– *61+ tilni qo‘llab-quvvatlash;*
– *16 til uchun 46 ta statistik modellar;*
– *oldindan o‘rgatilgan so‘z vektorlari;*
– *yuqori tezlikda tahlil qilish;*
– *“chuqur” o‘rganish bilan oson integratsiya;*
– *POS teglashtirish;*
– *o‘zaro bog‘liqliklarni tahlil qilish;*
– *sintaktik tahlil asosida gaplarni bo‘laklarga ajratish;*
– *NER vizualizatorlari;*
– *satrni xesh bilan qulay moslashtirish;*
– *NumPy ma’lumotlar bazasiga eksport qilish;*
– *samarali binar (ikkilik) serializatsiya;*
– *til modellarini qulay va sodda tahrirlash;*
– *ishonchli va yuqori aniqlikdagi tahlil.*

Ushbu maqolada quyidagilar ko‘rib chiqiladi:

– *NLPdagi asosiy atama va tushunchalar;*
– *ushbu tushunchalarni spaCyda qanday amalga oshirish;*
– *spaCyda o‘rnatilgan funksiyalarni qanday sozlash va kengaytirish;*
– *matn bo‘yicha asosiy statistik tahlilni qanday amalga oshirish;*
– *strukturlanmagan matnni qayta ishlash uchun pipeline jarayonini amalga oshirish.*

SpaCy har xil turdagi modellarga ega bo‘lib, ingliz tili uchun standart model *en_core_web_sm* hisoblanadi. Ingliz tilida modellar va ma’lumotlarni yuklab olish uchun quyidagi dastur kodidan foydalaniladi:

python -m spacy download en_core_web_sm

Python muhitida spaCy moduli quyidagicha yuklab olinadi:

```
import spacy
nlp = spacy.load(‘en_core_web_sm’)
```

Bu yerda NLP obyekti til modeli namunasidir. Ushbu maqolada NLP obyekti *en_core_web_sm* orqali yuklangan til modeliga ishora qiladi.

SpaCydagi asosiy obyektlar Doc va Vocab hisoblanadi. Doc obyekti tokenlar ketma-ketligi va ularning barcha izohlarini saqlaydi. Vocab obyekti barcha hujjatlar uchun umumiy ma’lumotlar taqdim etadigan yordamchi jadvallar to‘plamini saqlaydi.

Satrlar, so‘z vektorlari va *leksik atributlar*ni markazlashtirilgan tarzda saqlash orqali ushbu ma’lumotlarning bir nechta nusxasini saqlashning oldi olinadi.

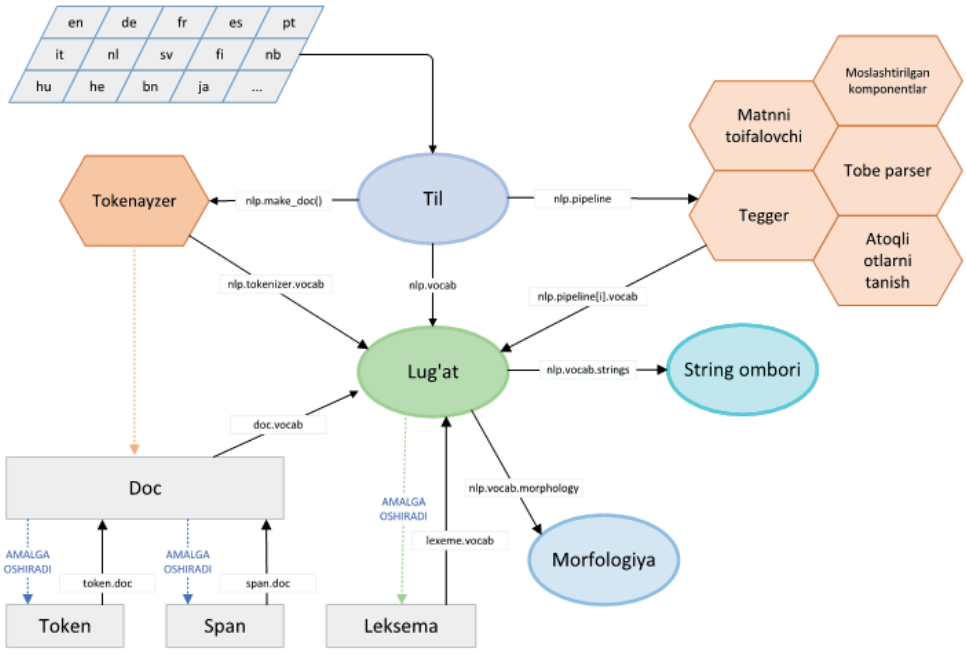
Matn izohlari ma’lumotlarning yagona manbasini ta’minlash uchun mo‘ljallangan bo‘lib, *Doc* obyekti ma’lumotlarni boshqarish, *Span* va *Token* esa unga ishora qiluvchi tasvirlardir. *Doc* obyekti *Tokenizer* obyekti tomonidan yaratiladi va keyin *pipeline* komponentlari tomonidan *in-place* o‘zgartiriladi.

1-jadval

NLP kutubxonalari

TOOL lar	XUSUSIYATLAR
NLTK	• Eng mashhur va to‘liq NLP kutubxonasi. • Har bir NLP vazifasiga turlicha yondashuv. • Ko‘plab tillarni qo‘llab-quvvatlash. • Integratsiyalanmagan so‘z vektorlari.
spaCy	• Eng tezkor NLP freymvorki. • Har bir vazifaga yagona yuqori optimallashtirilgan qulaylik qo‘shilganligi bois uni oson o‘rganish mumkinligi. • Ba’zi modellarni o‘rganish uchun neyron tarmoqlarni qo‘llab-quvvatlaydi . • Hamma tilga muvofiq kelavermaydi.
scikit learn	• Mashinali ta’limni amalga oshirish uchun eng samarali. • Matnni qayta ishlash uchun neyron tarmog‘i mavjud emas. • Ko‘plab tillarni qo‘llab-quvvatlaydi
gensim	• Katta ma’lumotlar to‘plamlari bilan ishlaydi va ma’lumotlar oqimini qayta ishlaydi • Deep Learning`ni qo‘llab-quvvatlaydi • Asosan, klassifikatsiyalanmagan matnni modellashtirishga mo‘ljallangan

Language obyekti ushbu komponentlarni muvofiqlashtirib, boshlang‘ich matnni qabul qiladi va izohli hujjatni *pipeline* orqali qaytaradi (3-rasm) [18, 575–584-b.].

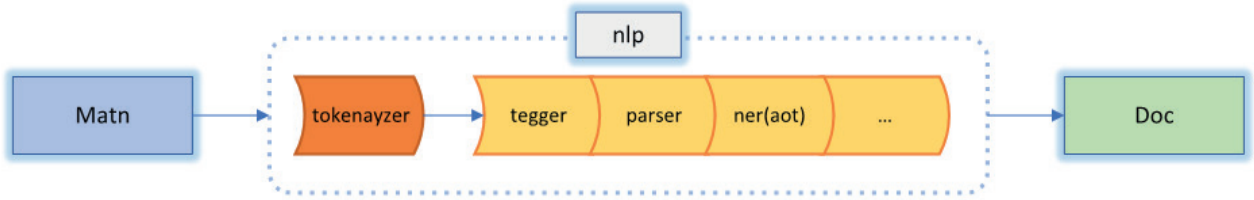


3-rasm. SpaCy arxitekturasi

ISSN 2181-9637
4 / 2022

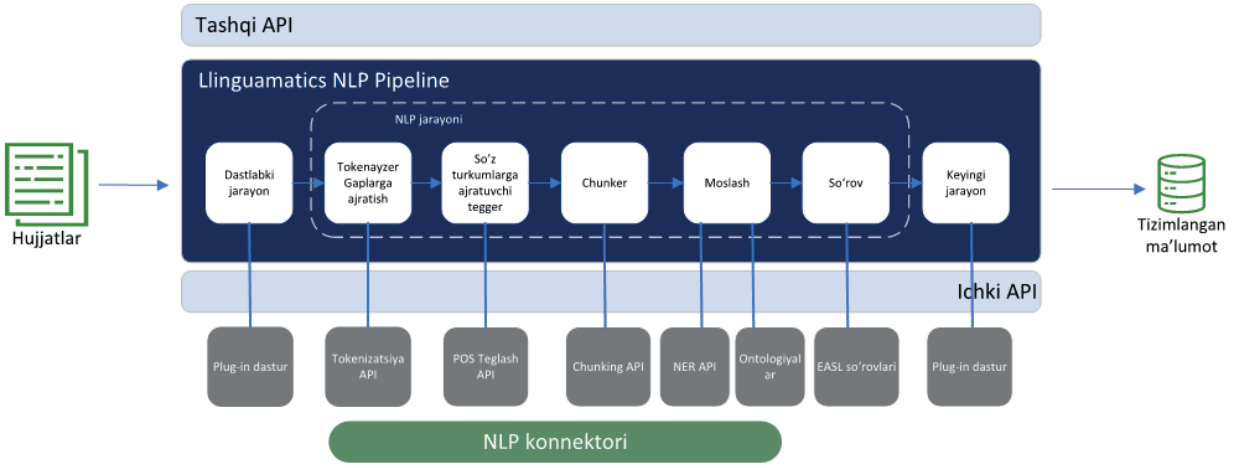
Pipeline jarayoni

Matnda NLP metodi chaqirilganda, spaCy ushbu matnni tokenlarga ajratib, Doc obyektini shakllantiradi. So‘ngra Doc pipeline deb nomlanuvchi ketma-ket bir necha bosqichda qayta ishlanadi. Pipeline standart modellarda quyidagi komponentlardan iborat: *tegger, parser* va *entity recognizer*. Har bir komponent parametr sifatida Doc obyektini oladi va uning kengaytirilgan (har bir token yoki butun Doc uchun yangi atributlar bilan) versiyasini qaytaradi (4-rasm) [13, 411–420-b.; 15, 25–36-b.].



4-rasm. Pipeline jarayoni

Pipeline til modeliga bog‘liq bo‘lib, uning asosiy versiyasi modelning metama’lumotlarida aniqlanadi. Pipeline arxitekturasi quyidagi 5-rasmda keltirilgan [19, 193–200-b.].



5-rasm. Pipline arxitekturasi

Quyidagi misolda berilgan matnni tokenlarga ajratish jarayoni keltirilgan:

```
import spacy
nlp = spacy.load(“en_core_web_sm”)
introduction_text = (‘This tutorial is about Natural Language Processing in Spacy.’)
introduction_doc = nlp(introduction_text)
# Berilgan hujjat uchun tokenlarni chiqarib olish
print ([token.text for token in introduction_doc])
```

Gaplarni aniqlash – berilgan matndagi gaplarning boshi va oxirini aniqlash jarayoni bo‘lib, bu bosqichda matnni lingvistik ma’noli birliklarga bo‘lish amalga oshiriladi. Matnni qayta ishlashda ushbu birliklardan *so‘z turkumlariga ajratish* (part of speech tagging) va *obyektni ajratib olish* (entity extraction) kabi vazifalarda foydalaniladi. SpaCyda sents xususiyati gaplarni aniqlash uchun ishlatiladi:

```
import spacy
nlp = spacy.load(“en_core_web_sm”)
introduction_text = (‘Jhonn Terry is a
```

Python developer currently working for a London-based Alpha company. He is interested in learning Natural Language Processing.')

```
about_doc = nlp(introduction_text)
sentences = list(about_doc.sents)
len(sentences)
for sentence in sentences:
  print (sentence)
```

Pipline komponentlari quyidagi jadvalda keltirilgan (2-jadval).

2-jadval

Pip komponentlar jadvali

Qiymat	Komponent	Izoh
Tagger	Tagger	Tokenlarga mansub so‘z turkumlarini aniqlash (part-of-speech-tagging)
Parser	DependencyParser	Tokenlarning o‘zaro bog‘liqligini o‘rnatish (dependency labels)
Ner	EntityRecognizer	Matndagi nomlangan obyektlarni aniqlash (named entities)
entity_linker	EntityLinker	Bilimlar bazasidagi obyektlarga ID berish (obyektlarni bog‘lash)
Textcat	TextCategorizer	Matnga kategoriya berish
entity_ruler	EntityRuler	Shablon bo‘yicha topilgan nomli obyektlarni matnga tayinlash (pattern rules)
Sentencizer	Sentencizer	Tokenlar orasidagi bog‘liqlikni topmasdan, qoidaga asoslangan jumlalarni ajratish
merge_noun_chunks	merge_noun_chunks	Ismlar ketma-ketligini bitta tokenga birlashtiradi. Tagger va parserdan so‘ng amalga oshiriladi
merge_entities	merge_entities	Barcha obyektlarni bitta tokenga birlashtiradi. Nerdan keyin amalga oshirilishi lozim
merge_subtokens	merge_subtokens	Parser tomonidan qabul qilingan subtokenslarni bitta tokenga birlashtiradi. Parserdan keyin amalga oshirilishi lozim

Tokenizatsiya – bu matnni tokenlar deb ataladigan mazmunli segmentlarga ajratish jarayonidir. Tokenizatorga parametr sifatida *Unicode* kodlash tizimidagi matn qabul qilinib, chiqishda Doc obyekti hosil bo‘ladi. So‘z tokenlari har qanday NLP masalasida qatnashadigan matnning asosiy birliklari hisoblanadi. Matnni qayta ishlashda birinchi qadam uni tokenlarga ajratishdir. Quyida keltirilgan kodda ushbu sinfning NLP obyektini yaratish uchun Spacy til sinfini import qilish zarur. SpaCy da tokenizatsiya jarayoni orqali hosil qilingan elementlardan boshlang‘ich matnni qayta hosil qilish mumkin. Ushbu skript matndagi tokenlar sonini hisoblaydi [2, 76–89-b.; 3, 143–150-b.; 18, 575–584-b.; 20, 695–709-b.]:

```
import spacy
nlp = spacy.load("en_core_web_sm")
introduction_text = ('Jhonn Terry is a Python developer currently working for a London-based Alpha company. He is interested in learning Natural Language Processing.')
doc = nlp(introduction_text)
vocab = {}
for token in doc:
  if token.text not in vocab.keys():
   vocab[token.text] = 1
  else:
   vocab[token.text] += 1
print(vocab)
```

SpaCyda tokenizatsiya uchun maxsus qoidalar qo‘shish va o‘z tokenizeringizni yaratish imkoni mavjud. Tokenizatsiya matnni mazmunli birliklarga ajratadi. Ushbu birliklar keyingi qadamlarda tahlil qilish uchun ishlatiladi.

So‘z turkumlariga ajratish (Parts of speech, POS)

Tokenizatsiya jarayonidan so‘ng Doc obyektidagi kontekstda qaysi teg yoki yorliq qo‘llanilishini taxmin qilish imkonini beruvchi statistik modeldan foydalaniladi

ISSN 2181-9637
4 / 2022

[21, 309–323-b.]. Til grammatikasi o‘rganilganda, *ot, fe’l, sifat* va *qo‘shimchalar* o‘rtasidagi farq tushuniladi. Bu tabiiy tilni qayta ishlashning muhim elementi hisoblanadi. SpaCy matnni so‘zlar ro‘yxatiga ajratadi. So‘ngra har bir so‘zni kontekst asosida so‘z turkumiga ajratish uchun qulay vositalarni taqdim etadi. Lingvistik izohlar token obyektining atributlari sifatida aniqlanadi. Ko‘pgina NLP kutubxonalari singari spaCy ham xotiradan foydalanishni kamaytirish va samaradorlikni oshirish uchun barcha satrlarni xesh qiymatlariga kodlaydi. Shunday qilib, atributning o‘qiladigan satrli atributini olish uchun uning nomiga pastki chiziq “_” qo‘shilishi kerak:

```
import spacy
nlp = spacy.load(“en_core_web_sm”)
introduction_text = (‘Jhonn Terry is a Python developer currently working for a London-based Alpha company. He is interested in learning Natural Language Processing.’)
doc = nlp(introduction_text)
for token in doc:
  print(token.text, token.lemma_, token.pos_, token.tag_, token.dep_,
        token.shape_, token.is_alpha, token.is_stop, token.idx)
```

Yuqoridagi skript tokenlar haqida quyidagi ma’lumotlar chiqadi (3-jadval).

3-jadval

Skriptning tokenlar haqidagi ma’lumotlari

TEXT	LEMMA	POS	TAG	DEP	SHAPE	ALPHA	STOP
O‘zgarishsiz token	Normal shakl	Coarse part-of-speech	Fine-grained part-of-speech	Bo‘gliqlilik haqida ma’lumot	Tokenning umumlashgan shakli (orfografik belgilarni ifodalaydi)	Token harflardan iboratmi?	STOP so‘zmi?

Shu bilan birga, tokenlarda maxsus belgilar o‘rnatish uchun tokenizatsiya jarayonini sozlash mumkin. Bu ko‘pincha defis bilan birlashtirilgan so‘zlar uchun qo‘llaniladi. SpaCy NLP obyektidagi tokenizator xususiyatini yangilash orqali tokenizatsiyani sozlash imkonini beradi:

```
import re
import spacy
from spacy.tokenizer import Tokenizer
custom_nlp = spacy.load(“en_core_web_sm”)

introduction_text = (‘Jhonn Terry is a Python developer currently working for a London-based Alpha company. He is interested in learning Natural Language Processing.’)
prefix_re = spacy.util.compile_prefix_regex(custom_nlp.Defaults.prefixes)
suffix_re = spacy.util.compile_suffix_regex(custom_nlp.Defaults.suffixes)
infix_re = re.compile(r’’’[-~]’’’)

def customize_tokenizer(nlp):
    # Adds support to use `-` as the delimiter for tokenization
    return Tokenizer(nlp.vocab, prefix_search=prefix_re.search,
                     suffix_search=suffix_re.search,
                     infix_finditer=infix_re.finditer,
                     token_match=None
                    )
custom_nlp.tokenizer = customize_tokenizer(custom_nlp)
custom_tokenizer_about_doc = custom_nlp(introduction_text)
print([token.text for token in custom_tokenizer_about_doc])
```

Stop Words. Stop Words tilda eng keng tarqalgan so‘zlardir. Ingliz tilida Stop Words sifatida the, are, but, they kabi so‘zlar e’tiborga olinadi. Ko‘pgina gaplar mantiqiy to‘liq jumlalar bo‘lishi va Stop Words`ni o‘z ichiga olishi kerak. Odatda, matnlarni tahlil qilishda Stop Words matndan olib tashlanadi. Chunki ular ahamiyatsiz va so‘z chastotasi tahlilini buzadi. SpaCyda ingliz tili uchun Stop Words ro‘yxati mavjud.

Lemmatizatsiya. Lemmatizatsiya – so‘zning normal shaklini topish, fleksiyaga uchragan so‘z-shakl asosini tiklash, shu bilan birga, reduksiyalangan shaklning tilga tegishli bo‘lishini ta’minlash jarayoni. Normal shakl yoki asos lemma deb ataladi. Lemmatizatsiya matndagi so‘zlarning flektiv shakllarini bitta element sifatida tahlil qilish imkoniyatini taqdim etadi. Shuningdek, u matnni normallashtirishga yordam beradi.

```
import spacy
nlp = spacy.load(“en_core_web_sm”)
introduction_text = (‘Jhonn Terry is helping organize a developer conference on Applications of Natural Language Processing. He keeps organizing local Python meetups and several internal talks at his workplace.’)
conference_help_doc = nlp(introduction_text)
for token in conference_help_doc:
  print (token, token.lemma_)
```

Ushbu misolda *tashkil qilish (organizing)* so‘z birikmasi o‘zining lemma shakliga qisqaradi. Agar matnni lemmatizatsiya qilmasangiz, unda *tartibga solish (organize)* va *tartibga solish* (organizing) – har ikkalasi ham o‘xshash ma’noga ega bo‘lsa ham, turli belgilar hisoblanadi. Lemmatizatsiya o‘xshash ma’noga ega bo‘lgan takroriy so‘zlardan qochishga yordam beradi.

So‘z chastotasi. Keyingi qadamda berilgan matnni tokenlarga ajratish va uning ustida statistik tahlilni amalga oshirish mumkin. Ushbu tahlil matndagi umumiy so‘zlar yoki unikal so‘zlar kabi ma’lumotlarni berishi mumkin:

```
import spacy
from collections import Counter
nlp = spacy.load(“en_core_web_sm”)
complete_text = (‘First, I wake up. Then, I get dressed. I walk to school.’
+ ‘I do not ride a bike. I do not ride the bus. I like to go to school. ‘
+ ‘It rains. I do not like rain. I eat lunch. I eat a sandwich and an apple.’
+ ‘I play outside. I like to play. I read a book. I like to read books.’
+ ‘I walk home. I do not like walking home. My mother cooks soup for dinner. ‘
+ ‘The soup is hot. Then, I go to bed. I do not like to go to bed.’)

complete_doc = nlp(complete_text)
# Stop words va tinish belgilarini olib tashlash
words = [token.text for token in complete_doc
      if not token.is_stop and not token.is_punct]
word_freq = Counter(words)

# Ko‘p uchraydigan so‘zlar (chastotasi bilan)
common_words = word_freq.most_common(5)
print (common_words)

# Unikal so‘zlar
unique_words = [word for (word, freq) in word_freq.items() if freq == 1]
print (unique_words)
```

[(‘like’, 6), (‘walk’, 2), (‘school’, 2), (‘ride’, 2), (‘eat’, 2)]

[‘wake’, ‘dressed’, ‘bike’, ‘bus’, ‘rains’, ‘rain’, ‘lunch’, ‘sandwich’, ‘apple’, ‘outside’, ‘book’, ‘books’, ‘walking’, ‘mother’, ‘cooks’, ‘dinner’, ‘hot’]

Shunday qilib, har qanday strukturlanmagan matn mazmunini aniqlash uchun uni statistik tahlil qilish mumkin. Quyidagi misolda *Stop words* bilan birgalikdagi tahlil keltirilgan:

```
words_all = [token.text for token in complete_doc if not token.is_punct]
word_freq_all = Counter(words_all)

# Ko‘p uchraydigan so‘zlar (chastotasi bilan)
common_words_all = word_freq_all.most_common(5)
```

```
print (common_words_all)

[('I', 17), ('to', 8), ('like', 6), ('do', 5), ('not', 5)]
```

Statistik tahlil natijasiga ko‘ra, matndagi eng ko‘p uchragan beshta so‘zdan to‘rttasi *Stop Words* bo‘lib, ular matn haqida ko‘p ma’lumot bermaydi. Agar so‘z chastotasini tahlil qilishda *Stop Words*`lar hisobga olinsa, matn mazmuni noto‘g‘ri tahlil qilinadi. Shuning uchun *Stop Words*`larni tahlil jarayonida olib tashlash juda muhimdir.

Tadqiqot natijalari tahlili

POS tegging jarayonida matndagi har bir so‘zning qanday ishlatilishini tushuntiruvchi grammatik xususiyati aniqlanadi. Ingliz tilida 8 ta so‘z turkumi mavjud:

1. *Noun.*
2. *Pronoun.*
3. *Adjective.*
4. *Verb.*
5. *Adverb.*
6. *Preposition.*
7. *Conjunction.*
8. *Interjection.*

SpaCyda POS teglari Token obyektida atribut sifatida aniqlangan:

– tag_lists so‘z turkumlarining kengaytirilgan guruhini aniqlaydi;

– pos_lists so‘z turkumlarining oddiy guruhini aniqlaydi.

spacy.explain atributi orqali muayyan POS teg haqidagi tafsilotlarni olish mumkin. Quyidagi misolda POS teglaridan foydalanib, muayyan so‘z turkumiga mansub so‘zlarni ajratib olish mumkin:

```
import spacy
from collections import Counter
nlp = spacy.load("en_core_web_sm")
complete_text = ('First, I wake up. Then, I get dressed. I walk to school.'
+ 'I do not ride a bike. I do not ride the bus. I like to go to school. '
+ 'It rains. I do not like rain. I eat lunch. I eat a sandwich and an apple.'
+ 'I play outside. I like to play. I read a book. I like to read books.'
+ 'I walk home. I do not like walking home. My mother cooks soup for dinner. '
+ 'The soup is hot. Then, I go to bed. I do not like to go to bed.')
about_doc = nlp(complete_text)
nouns = []
adjectives = []
for token in about_doc:
  if token.pos_ == 'NOUN':
    nouns.append(token)
  if token.pos_ == 'ADJ':
    adjectives.append(token)
print(nouns)
print(adjectives)
```

Bog‘liqlikni tahlil qilish. spaCy bog‘liqlikni tahlil qiluvchi tez va aniq sintaktik analizatorga ega bo‘lib, unda iyerarxik tuzilmali amallarni bajaruvchi API interfeysi mavjud. Sintaktik analizator gap chegaralarini aniqlashni ta’minlaydi va asosiy so‘z birikmalari (chunks)ni aniqlab beradi. Mantiqiy qiymatni qaytaruvchi doc.is_parsed atributi yordamida Doc obyekti tahlil qilinganligini aniqlash mumkin.

```
import spacy
nlp = spacy.load("en_core_web_sm")
doc = nlp("Autonomous cars shift insurance liability toward manufacturers")
for chunk in doc.noun_chunks:
  print(f"{chunk.text}, {chunk.root.text}, {chunk.root.dep_}, {chunk.root.head.text}")
```

Natija:

```
Autonomous cars, cars, nsubj, shift
insurance liability, liability, dobj, shift
man\ufacturers, manufacturers, pobj, toward
```

Yuqoridagi skript qismlar haqida quyidagi ma’lumotlarni chiqaradi:

EXT	ROOT.TEXT	ROOT.DEP_	ROOT.HEAD.TEXT
So‘z birikmasi	Kalit so‘z	Sintaktik munosabat turi	Root ga nisbatan bosh so‘z

SpaCy bog‘liqlik daraxtidagi o‘zaro aloqaga ega so‘zlar uchun head va child atamalaridan foydalanadi.

Bog‘liqlik daraxti bo‘ylab harakatlanish

```
import spacy
nlp = spacy.load(“en_core_web_sm”)
doc = nlp(“Autonomous cars shift insurance liability toward manufacturers”)
for token in doc:
    print(token.text, token.dep_, token.head.text, token.head.pos_,
        [child for child in token.children])
```

Formatlangan natija

4-jadval

Dasturning formatlangan natijalari

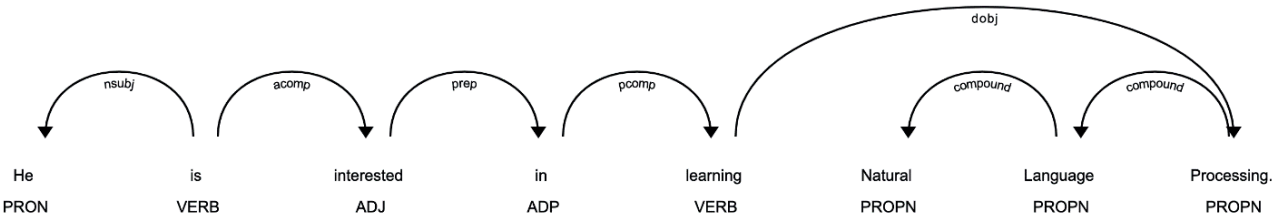
TEXT	DEP	HEAD TEXT	HEAD POS	CHILDREN
Autonomous	Amod	Cars	NOUN	
Cars	Nsubj	Shift	VERB	Autonomous
Shift	ROOT	Shift	VERB	cars, liability, toward
Insurance	Compound	liability	NOUN	
Liability	Dobj	Shift	VERB	Insurance
Toward	Prep	Shift	NOUN	Manufacturers
manufacturers	Pobj	toward	ADP	

Sintaktik tobeliklar daraxt shaklida ifodalanganligi uchun har bir so‘z faqat bitta head`ga ega.

Vizualizatsiya. SpaCy displaCy deb nomlangan ichki vizualizator mavjud bo‘lib, undan brauzer yoki Jupyter daftaridagi so‘zlarning o‘zaro bog‘liqligi yoki nomlangan obyektlarni ko‘rish uchun foydalanishingiz mumkin:

```
import spacy
from spacy import displacy
nlp = spacy.load(‘en_core_web_sm’)
about_interest_text = (‘He is interested in learning Natural Language Processing.’)
about_interest_doc = nlp(about_interest_text)
displacy.serve(about_interest_doc, style=’dep’)
```

Yuqoridagi kod oddiy veb-serverni ishga tushiradi. Brauzerda http://127.0.0.1:5000 havolani ochish orqali vizualizatsiyani ko‘rishingiz mumkin (6-rasm).



6-rasm. POS vizualizatsiyasi

Nomlangan obyekt (Named Entity Recognition, NER). Nomlangan obyekt “haqiqiy obyekt” bo‘lib, unga *shaxs, mamlakat, mahsulot yoki kitob nomi* kabi nom berilgan. SpaCy hujjatdagi turli nomdagi obyektlarni model orqali taniy oladi. Modellar statistik bo‘lib, ular o‘qitilgan misollarga juda bog‘liq bo‘lgani sababli bu har doim ham mukammal ishlamaydi va foydalanish holatlaringizga qarab, qo‘shimcha amallarni talab qilishi mumkin.

Obyekt izohlariga murojaat qilishning standart usuli *doc.ents* bo‘lib, u yerda *Span* obyektlari ketma-ketligi saqlanadi. Obyekt turi xesh qiymati sifatida yoki ent.label va ent.label_ atributlari yordamida satr sifatida aniqlangan. Span obyekti – bu tokenlar ketma-ketligi hisoblanadi. Token obyektining izohlariga token.ent_iob (obyektning boshi, o‘rtasi yoki oxiri) va token.ent_type (obyekt turi, agar qiymat o‘rnatilmagan bo‘lsa, bo‘sh

qator qaytariladi) atributlari yordamida ham murojaat qilish mumkin.

```
import spacy
nlp = spacy.load("en_core_web_sm")
# ner – pipline jarayoni bir qismidir.
doc = nlp("San Francisco considers banning sidewalk delivery robots")
# hujjat darajasi
ents = [(e.text, e.start_char, e.end_char, e.label_) for e in doc.ents]
print("Obyekt: ", ents)
# token darajasi
ent_san = [doc[0].text, doc[0].ent_iob_, doc[0].ent_type_]
ent_francisco = [doc[1].text, doc[1].ent_iob_, doc[1].ent_type_]
print("'Sun' tokeni: ", ent_san)
print("'Francisco' tokeni: ", ent_francisco)
Natija:
[('San Francisco', 0, 13, 'GPE')]
['San', 'B', 'GPE']
['Francisco', 'I', 'GPE']
```

Xulosalar

Ushbu maqolada ko'pgina sohalarda qo'llanilishi mumkin bo'lgan NLPning keng ko'lamli vositalari ko'rib chiqildi. Inson va mashinali ta'lim jarayoni turlicha bo'lganligi sababli kompyuterda tabiiy tilni qayta ishlashning bir qator usullaridan foydalaniladi. SpaCy, NLTK kabi Python kutubxonalari ish jarayonini osonlashtiradi. Bugungi kunda SpaCy *ishonchli* va *ommabop* Python kutubxonasi hisoblanib, *tezligi, foydalanish qulayligi, aniqligi* va *moslashuvchanligi* tufayli NLP ilovalarida undan keng miqyosda foydalanilmoqda. Har bir daqiqada, asosan, matnli ma'lumotlar turli formatlarda yaratiladi, masalan: *SMS, sharhlar, elektron pochta xabarlari* va boshqalar. Ushbu maqolada keltirilgan usullar orqali SpaCy kutubxonasidan foydalangan holda *pipeline* jarayoni amalga oshirildi. *Pipeline* jarayoni orqali matnni *tegger, parser* va *entity recognizer* atributlari qiymatlari shakllantirildi. Mazkur bosqichda strukturlanmagan matndan ma'lumot olish, "nomlangan obyektlar"ni aniqlash, matndagi so'z birliklarini tahlil qilish amalga oshiriladi. NLP texnologiyalaridan foydalanish axborot tizimlarini ishlab chiqishda biznes-jarayonlarni modellashtirish va ish samaradorligini oshirishni sezilarli darajada yaxshilaydi. Shuningdek, SpaCy paketida o'zbek tili modelini shakllantirish natijasida quyidagi NLP masalalarini hal qilish mumkin:

– *o'zbek tili milliy korpusini yaratish;*

– *o'zbek tili morfologik analizatorini ishlab chiqish;*

– *o'zbek tili sintaktik analizatorini ishlab chiqish;*

– *o'zbek tili semantik analizatorini ishlab chiqish.*

REFERENCES

1. GPT-3 Powers the NextGeneration of Apps. Available at: https://openai.com/blog/gpt-3-apps/.

2. Bol'shakova Ye.I., Vorontsov K.V., Yefremova N.E., Klyshinskiy E.S., Lukashevich N.V., Sapin A.S. Avtomaticheskaya obrabotka tekstov na yestestvennom yazyke i analiz dannykh [Automatic natural language processing and data analysis]. Moscow, NIU VShE Publ., 2017, 269 p.

3. Kharis M., Laksono K., Suhartono, Ridwan A., Mintowati, Yuniseffendri. Tokenization and lemmatization on German learning textbook level A1 of CEFR Standard. *Journal of Higher Education Theory and Practice*, 2022, no. 22 (1). DOI: 10.33423/jhetp.v22i1.4971/.

4. Chantrapornchai C., Tunsakul A. Information extraction on tourism domain using SpaCy and BERT. *ECTI Transactions on Computer and Information Technology*, 2021, 15 (1). DOI: 10.37936/ecticit.2021151.228621/.

5. Yanti R.M., Santoso I., Suadaa L.H. Application of named entity recognition via Twitter on SpaCy in Indonesian. Case Study: power failure in the special region of Yogyakarta. *Indonesian Journal of Information Systems*, 2021. DOI: 10.24002/ijis.v4i1.4677/.

6. Kharis M., Laksono K., Suhartono, Ridwan A., Mintowati, Yuniseffendri. Tokenization and lemmatization on german learning textbook level A1 of CEFR Standard. *Journal of Higher Education Theory and Practice*, 2022, no. 22 (1). DOI: 10.33423/jhetp.v22i1.4971/.

7. Cing D.L., Soe K.M. Improving accuracy of part-of-speech (POS) tagging using hidden markov model and morphological analysis for Myanmar language. *International Journal of Electrical and Computer Engineering*, 2020, no. 10 (2). DOI: 10.11591/ijece.v10i2. pp2023-2030/.

8. Chandola D., Garg A., Maurya A., Kushwaha A. Online Resume Parsing System Using Text Analytics, 2015. Available at: http://www.jmdet.com/wp-content/uploads/2015/08/CR9.pdf/.

9. Turgunbaev R., Elov B. The use of machine learning methods in the automatic extraction of metadata from academic articles. *International Journal of Innovations in Engineering Research and Technology*, 2021, no. 8 (12), pp. 72-79. DOI: 10.17605/OSF.IO/QB5PZ/.

10. Elov B., Akhmedova Kh. A mathematical model that semantically analyzes polysemantic words. *Journal of Pedagogical Inventions and Practices*, 2021, no. 3, pp. 119-122. Available at: https://zienjournals.com/index.php/jpip/article/view/469/.

11. Jabeen H. Stemming and lemmatization in Python. *Towardsdatascience,* 2018.

12. Chong C., Sheikh U.U., Samah N.A., Sha'Ameri A.Z. Analysis on reflective writing using natural language processing and sentiment analysis. *IOP Conference Series: Materials Science and Engineering*, 2020, no. 884 (1). DOI: 10.1088/1757-899X/884/1/012069/.

13. Honnibal M., Montani I. SpaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *Appear*, 2017, no. 7 (1), pp. 411-420. Available at: https://sentometrics-research.com/publication/72/.

14. Shelar H., Kaur G., Heda N., Agrawal P. Named entity recognition approaches and their comparison for custom NER Model. *Science and Technology Libraries*, 2020, no. 39 (3), pp. 324-337. DOI: 10.1080/0194262X.2020.1759479/.

15. Jugran S., Kumar A., Tyagi B.S., Anand V. Extractive automatic text summarization using SpaCy in Python NLP. 2021 International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE, 2021. DOI: 10.1109/ICACITE51222.2021.9404712/.

16. Honnibal M. Founder and CTO, SpaCy.io. Available at: http://scholar.google.com/citations?user=FXwlnmAAAAAJ&hl=en/.

17. Ines, a software developer working on Artificial Intelligence and Natural Language Processing technologies, and the co-founder and CEO of Explosion. Available at: https://ines.io/.

18. Saloot M. A., Pham D.N. Real-time Text Stream Processing: A Dynamic and Distributed NLP Pipeline. ACM International Conference Proceeding Series. 2021. DOI: 10.1145/3459104.3459198/.

19. Rai A., Borah S. Study of various methods for tokenization. *Lecture Notes in Networks and Systems,* 2021, vol. 137. DOI: 10.1007/978-981-15-6198-6_18/.

20. Pudasaini S., Shakya S., Lamichhane S., Adhikari S., Tamang A., Adhikari S. Application of NLP for information extraction from unstructured documents. *Lecture Notes in Networks and Systems*, 2022, vol. 209. DOI: 10.1007/978-981-16-2126-0_54/.

21. Pota M., Marulli F., Esposito M., de Pietro G., Fujita H. Multilingual POS tagging by a composite deep architecture based on character-level features and on-the-fly enriched Word Embeddings. *Knowledge-Based Systems*, 2019, vol. 164. DOI: 10.1016/j.knosys.2018.11.003/.

22. Kumar A., Katiyar V., Kumar P. A Comparative analysis of pre-processing time in summary of hindi language using Stanza and Spacy. IOP Conference Series: Materials Science and Engineering, 2021, no. 1110 (1). DOI: 10.1088/1757-899x/1110/1/012019/.

Taqrizchi:

Mo'minov B.B., t.f.d., prof., TATU Aborot texnologiyalarining dasturiy ta'minoti kafedrasi mudiri.