

TABIY TILNI QAYTA ISHLASH (NLP)

Morfologik analizatorni yaratish usullari

Botir Elov¹
Shahlo Hamroyeva²
Dilrabo Elova³

Abstrakt

Morfologik tahlil jarayoni imloni tuzatish vositalari, parserlar, mashina tarjima tizimlari va elektron lug'atlar kabi tabiiy tilni qayta ishlash tizimlarining muhim tarkibiy qismidir. Ushbu maqolada matn analizatori bosqichlari, morfologik analizator va morfologik generator yaratish usullari tavsiflangan. Morfologik analizatorni yaratishda Python tilidagi NLTK paketi vositalaridan foydalanish yo'llari, dasturiy kodlarning namunalari keltirilgan. Shuningdek, morfologik tahlil jarayoni boshqichlari (*fleksiya, hosila, affikspidlarni aniqlash, birikma shakllari*) asosida morfologik analizator sturkturasi va arxitekturasini taqdim etilgan.

Kalit so'zlar: *Tabiiy tilni qayta ishlash, NLP, Python, morfologik analizator, token, lemmatizatsiya, stemming, morfologik generator, qidiruv tizimi, stemming algoritmi, PorterStemmer.*

Kirish

Morfologiya – morfema vositasida so'zlarning grammatik ma'nolarini o'rganuvchi bo'lim. *Morfema* ma'noli qismlarga bo'lin-

¹Elov Botir Boltayevich – Alisher Navoiy nomidagi Toshkent davlat o'zbek tili va adabiyoti universiteti – Kompyuter lingvistikasi va raqamli texnologiyalar kafedrasini mudiri, texnika fanlari bo'yicha PhD

E-pochta: elov@navoiy-uni.uz

ORCID: 0000-0001-5032-6648

²Hamroyeva Shahlo Mirdjonovna – Alisher Navoiy nomidagi Toshkent davlat o'zbek tili va adabiyoti universiteti, Amaliy filologiya kafedrasini filologiya fanlari doktori (DSc)

E-pochta: shaxlo.xamrayeva@navoiy-uni.uz

ORCID: 0000-0002-5429-4708

³Elova Dilrabo Qudratilloevna – Alisher Navoiy nomidagi Toshkent davlat o'zbek tili va adabiyoti universiteti, Amaliy filologiya kafedrasini o'qituvchisi

E-pochta: elovadilrabo@navoiy-uni.uz

ORCID: 0000-0002-2329-1811

Iqtibos uchun: Elov, B.B., Hamroyeva, Sh.M., Elova, D.Q. 2022. "Morfologik analizatorni yaratish usullari". *O'zbekiston: til va madaniyat. Amaliy filologiya* 1(5): 88-108.

maydigan tilning eng kichik birligidir. Ushbu maqolada ingliz tilidan tashqari tillar uchun stemming va lemmatizatsiya, stemmer va lemmatizator, mashinali o'rganish vositalari, qidiruv tizimlaridan foydalangan holda **morfologik analizator** va **morfologik generator** yaratish masalasi tahlil qilinadi.

Bugungi kunda jahonda bir qator olimlar morfologik analizatorini yaratish ustida ilmiy izlanishlar olib bormoqda. Jumladan, Adnan Öztürel, Tolga Kayadelen va Isin Demirsahin turk tili morfologiyasining keng qamrovli modelini va uni amalga oshiradigan ochiq manbali morfologik analizatorini taqdim etgan [Öztürel, Kayadelen, Demirşahin 2019]. Ushbu model turk tili morfologiyasi, sintaksisining nozik jihatlarini qamrab olgan, undan til modelini ishlab chiqishda yo'l-yo'riq sifatida foydalanish mumkin. Model turk tili morfotaktikasini OpenFst yordamida chekli holat transduserlari va morf fonemik jarayonlarni Thrax grammatikalari sifatida amalga oshiradi. Arab tilshunoslari Y.Jaafar va K.Bouzoubaa tomonidan arab morfologik analizatorlaridan sintaktik tahlil dasturlari, qidiruv tizimlari va mashina tarjimai tizimlarida foydalanish usullari keltirilgan.

Tabiiy tilni tushunish (Natural Language baholashga bag'ishlangan arab tili korpusining birinchi versiyasi yaratilgan [Jabbar, Iqbal, Akhunzada, Abbas 2018]. G.Szabó, L.Kovács tomonidan venger tili uchun eng keng tarqalgan va eng ilg'or to'rtta morfologik analizator (*Hunmorf-Okamorf*, *Hunmorf-Foma*, *Humor* va *Hunspell*) tahlil qilingan, taqqoslangan. Ular analizatorlarning lemmatizatsiya xususiyatlari o'rniga Abstraktning token tizimlari qiyoslangan [Szabó, Kovács 2018].

Jahon kompyuter lingvistikasida tabiiy tilga ishlov berish muammolari A.Keyler, K.V.Linden, N.Vard [Daniel, Jurafsky, James, Martin 2000], M.A.Mohri [Mohri] tomonidan yoritildi. Matnga avtomatik ishlov berish texnologiya va vositalari V.Axo Alfred, S.Lam Monika, S.Ravi, D.Ulman Djeffri [Ахо Альфред, Лам Моника, Сети Рави, Ульман Джеффри 2008], morfologik tahlilning nazariy asoslari S.Yu.Toldova, A.A.Bonch-Osmolovskaya [Толдова, Бонч-Осмоловская 2011], T.Sadikov [Садыков, Кочконбаева 2018.] ishlarida tavsiflandi. Rus kompyuter lingvistikasida O.V.Dereza, D.A.Kayutenko, A.S.Fenogenova, M.Hakimovning morfologik analizator, uning umumiy tavsifi, ishlash prinsiplari, tilga oid grammatik kategoriya, dastur uchun lingvistik modullar tavsifi haqidagi ishlarini ta'kidlash o'rinli.

Morfoanaliz xususiyatlari D.Rudolf [Rodolfe 2008],

morfoanalizning amaliy ahamiyati A.Ermakov [Ермаков], matnga avtomatik morfologik va sintaktik ishlov berish modeli va dasturlari I.M.Nojoy [Ножов 2003] tomonidan o'rganilgan.

Turkiy kompyuter lingvistikasida tabiiy tilning kompyuter ishlovi, jumladan, morfoanalizator tuzish masalalari atroflicha tadqiq etildi. Bunday ishlar sirasiga A.V.Dibo, A.V.Sheyimovich [Дыбо, Шеймович 2014], J.Suleymanov [Сулейманов, Гатиатуллин 1996], A.R.Gatiatullin, A.M.Bashirov [Сулейманов, Гильмуллин, Гатауллин 2017], P.V.Jeltov [Желтов 2017], N.A.Israilova, P.S.Bakasova [Исраилова, Бакасова 2017], N.A.Leontyev [Леонтьев 2018], V.V.Kukanova, A.Yu.Kadjiev [Куканова, Каджиев 2014] larning morfologik nalizator tuzish masalalariga bag'ishlangan tadqiqotlarini kiritish mumkin. Turkiy tillar morfologik tahlil tizimi hamda ularning modellari tavsifi, tahliliga bag'ishlangan tadqiqotlar ko'lamining kengligi ko'rsatadi. O'zbek tilidagi otlarni avtomatik tahlil qilish [Орхун 2018], so'z yasalishining formal modellari [Турсунов 2018] haqida qator maqolalar nashr etilgan. O'zbek tilida ot va sifat so'z turkumi morfologik analizini avtomatlashtirishning dasturiy tomonlari o'rganilgan [Болтаев, Ибрагимов 1992].

1. *NLPni ikkita asosiy komponentga ajratish Understanding, NLU).*

2. *Tabiiy tili generatsiya qilish (Natural Language Generation, NLG).*

NLP vazifalarini bajarishda talab etiladigan qadamlar ko'rib chiqiladi. Tabiiy tilning **manbaya nutq (tovush)** yoki **matn** bo'lishi mumkin. Matnni qayta ishlash bosqichlari 1-rasmda keltirilgan:



1-rasm. Matn analizatori bosqichlari

O'zbek kompyuter lingvistikasi sohasida haligacha **morfologik analizator** va **morfologik generator** yaratilmagan. Ushbu maqolada jahon tajribasidan kelib chiqqan holda ushbu masalali hal qilish usullarini ko'rib chiqamiz.

Morfologik analizator yaratishda quyidagilar tushunchalarni

o'rganish maqsadga muvofiq:

- 1) morfologik birliklar;
- 2) stemmer;
- 3) lemmatizatsiya;
- 4) o'zbek tilida stemmerni ishlab chiqish;
- 5) morfologik analizator;
- 6) morfologik generator;
- 7) qidiruv tizimini ishlab chiqish.

MORFOLOGIK BIRLIKLAR

Morfema tilning eng kichik ma'noli qismidir. Morfema ikki turga ajratiladi: **o'zak** morfema, qo'shimcha morfema (**affiks**) (rus.: суффиксы, префиксы, инфиксы и циркумфиксы). O'zaklar erkin morfemalar deb ham yuritiladi, chunki ulardan gaplarda affiks (**qo'shimcha**) qo'shmasdan ham foydalanish mumkin. Affikslar (**qo'shimcha**) tilda alohida qo'llanilmaydi. Chunki ular mustaqil shaklda lug'aviy ma'no anglatolmaydi va doimo mustaqil morfemalar bilan birga mavjud bo'ladi. Ingliz tilidagi **unbelievable** so'zini ko'rib chiqamiz. Bu yerda **believe** o'zak yoki erkin morfemadir. U mustaqil ravishda ma'no anglatadi; **un** va **able** morfemalari *affiks (qo'shimcha)* yoki *bog'langan morfemalar*dir. Ushbu morfemalar mustaqil shaklda mavjud bo'lolmaydi, ma'no anglatmaydi, ular o'zak bilan birga ishlatiladi [Gökçe 2014; Ruppenhofer, Steiner, Wiegand 2017]. O'zbek tilida *vatandoshlarimiz* so'zi **vatan** (asos)+**dosh** (yasovchi)+**lar** (ko'plik)+**imiz** (egalik) tarzida farqlanadi.

Tabiiy tillar grammatik shakl hosil qilishiga ko'ra quyidagi guruhlar ajratiladi:

amorf tillar (isolating languages) – maye xitoy, bamana, anam, Janubiy Sharqiy Osiyo mamlakatlari xalqlarining talay tillari [Minegishi 2011];

agglutinativ tillar (agglutinative languages) – turkiy, bantu, mo'g'ul, fin-ugor tillari [Gökçe 2014; Pan, Li, Yang, Dong 2020];

inflektiv tillar (inflecting languages) – hind-yevropa (lotin tili va b.) va som tillari *oilalari* [Hakkani-Tür, Oflazer, Tur, 2002];

polisintetik tillar – Shimoliy Amerika *indeyslari tillari* [Kelly, Wigglesworth, Nordlinger, Blythe 2014].

Amorf tillar. Bu tillarning lug'at tarkibi, asosan, bir bo'g'inli o'zaklardan iborat bo'lib, ularda *turlanish*, *tuslanish* xususiyatlari yo'q. Shu bois ushbu tillarda so'z yasovchi affikslar, yordamchi so'zlar vazifasini bajaruvchi yuklamalargina mavjud. Amorf tillarda so'zlar faqat mustaqil morfemalardan iborat bo'ladi; ular **zamon**

(o'tgan, hozirgi va kelajak) va **son** (birlik yoki ko'plik) ma'lumotlarini o'z ichiga olmaydi.

1. **Agglyutinativ tillar.** Agglyutinativ tillarda so'zlar o'zak va unga birikib keladigan qo'shimchalardan iborat bo'ladi; so'zning morfologik tarkibi (o'zak va qo'shimcha) aniq ajralib turadi. Bunda har bir qo'shimcha alohida ma'no, vazifa ifodalaydi. Masalan, turkiy tillarda, jumladan, o'zbek tilida yasama so'zlar va so'z shakllari asosga muayyan izchillik bilan qo'shimchalar qo'shish orqali hosil qilinadi. Bunda har bir qo'shimchaning o'z grammatik ma'nosi mavjud: **terimchilarimizga (ter-im-chi-lar-imiz-ga)**. Agglyutinativ tillarda murakkab ma'lumotni yetkazish uchun o'zak so'zlarning birikishi natijasida hosil qilinadigan so'zlardan foydalaniladi.

2. **Flektiv tillar** – qo'shimchalarning o'zak bilan birikib, unga singib ketishi bilan tavsiflanadi. Bunday tillarda grammatik ma'nolar fleksiya yo'li bilan ifodalanadi: mas., arab tilida kitob (birlik) – kutub (ko'plik). Rus tilida друг (birlik) – друзья (ko'plik). Flektiv tillarga hind-yevropa va som tillari oilalari kiradi. Flektiv tillar sintetik va analitik tillarga ajratiladi. Sintetik tillarda grammatik ma'nolar (gapda so'zlarning o'zaro munosabati) shakl yasovchi qo'shimchalar vositasida ifodalanadi (mas., rus, nemis tillari). Analitik tillarda esa grammatik ma'nolar so'z shakllari (shakl yasovchi qo'shimchalar) vositasida emas, balki yordamchi so'zlar, so'z tartibi, ohang yordamida ifodalanadi (mas., ingliz, fransuz, ispan tillari). Flektiv tillarda so'zlar sodda birliklarga bo'lingan bo'lib, bu sodda birliklarning barchasi turli xil ma'nolarni namoyon qiladi.

3. **Polisintetik tillar** – asosiy nutq birligi so'z-gapdir. Tasnif etilgan tillar orasiga qat'iy chegara qo'yib bo'lmaydi, chunki bir tilda uchraydigan ayrim til hodisalari boshqalarida ham uchrab qolishi mumkin. Masalan, Okeaniya tillarini ham amorf tillar, ham agglyutinativ tillar sifatida tavsiflash mumkin [Кузнецов 2010].

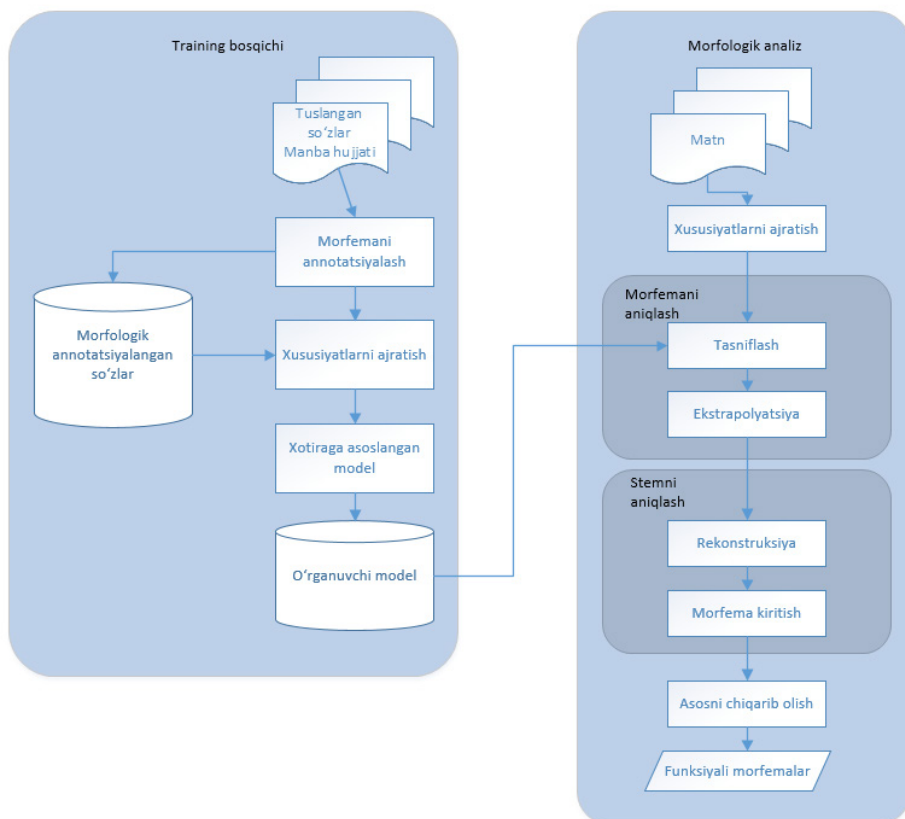
Tillarning morfologik (tipologik) tasnifi dunyo tillarini muayyan morfologik xususiyatlariga ko'ra guruhlash, ularning umumiy chizmasini yaratish ahamiyatli [Bareket, Tsarfaty 2021; Ledeneva, Sidorov, 2010; Pirkola 2001].

Morfologik jarayonlar quyidagi boshqichlarga bo'linadi [Jayan, Rajendran 2011]:

- 1) *fleksiya (inflection);*
- 2) *hosila (derivation);*
- 3) *yarim affikslarni aniqlash (semiaffixes);*

- 4) *birikma shakllari (combining forms)*;
- 5) *klitizatsiya (cliticization)*.

Shu bosqich asosida o'zbek tili morfologik analizatorining ishlash boshqichi strukturasi quyidagi 2-rasmda keltirilgan:



2-rasm. Morfologik analizator sturkturasi

Infleksiya bosqichida so'z *shaxs, son, zamon, jins* va boshqa grammatik kategoriyalarni ifodalovchi shaklga aylantiriladi. Ushbu bosqichda leksemaning sintaktik kategoriyasi o'zgarishsiz qoladi.

Derivatsion bosqichida so'zning sintaktik kategoriyasi o'zgaradi. Yarimaffikslarni aniqlash bosqichida qo'shma so'zlar, qisqartma shaklida morfemalar hosil qilinadi. Masalan: *noteworthy, antisocial, anticlockwise* va hokazo.

Stemmer. Stemmatizatsiya jarayonida so'zdan qo'shimchalarni olib tashlash orqali so'z o'zagini aniqlash amalga oshiriladi [Aslan, Gunal, Dincer 2018; Rifano, Fauzan, Makhi, Nadya, Nasikin, Putra, 2020; Yayla, Diyar Demirkol, Alqaraleh 2021]. Masalan, **tabiatdan** so'zi quyidagicha tahlil qilinadi: **[tabiat]** - dan (*Kelishik qo'shimchalari* $\in$ *Sintaktik shakl yasovchi*)

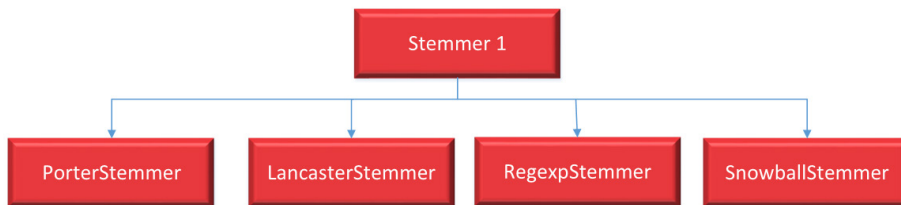
Qidiruv tizimlari (Google, Yahoo, Yandex) ma'lumot qidirishning aniqligini oshirish maqsadida asosan o'zakni aniqlash va ularni indekslangan so'zlar sifatida saqlashda **stemming**dan foydalaniladi. Qidiruv tizimlari bir xil ma'noga ega bo'lgan so'zlarni sinonimlar deb ataydi va bu so'rov natijalarni aniqroq amalga oshirishning bir turi bo'lishi mumkin.

Hozirgi kunda ko'plab morfoanalizatorlar Martin Porter tomonidan ishlab chiqilgan *stemming algoritmi*dan foydalanmoqda. Ushbu algoritm asosan ingliz tilidagi so'zlarda mavjud bo'lgan sufikslarni almashtirish va tahlil qilish uchun mo'ljallangan. NLTK paketida stemmingni amalga oshirish uchun **PorterStemmer** sinfining nusxasini yaratishimiz va keyin **stem()** metodini chaqirish orqali stemmingni amalga oshirishimiz mumkin (3-rasm). **NLTK**da PorterStemmer sinfidan foydalanib, stemming jarayonini ko'rib chiqamiz [Deepti Chopra, Jacob Perkins, Nitin Hardeniya 2016; Orhun, Tantug, Adali 2009]:

```
import nltk
from nltk.stem import PorterStemmer
stemmerporter = PorterStemmer()
print(stemmerporter.stem('removing'))
print(stemmerporter.stem('happies'))
```

remov
happi

PorterStemmer sinfi ingliz tilidagi ko'plab o'zaklar va so'z shakllari haqida ma'lumotga ega. O'zaklarni aniqlash jarayoni bir nechta bosqichdan iborat: so'zni *qisqaroq so'zga* yoki *o'zakka o'xshash ma'noga ega shaklga* aylantiradi. Stemmer I interfeysi **stem()** metodini aniqlaydi va barcha stemmerlar Stemmer I interfeysidan namuna qilib olingan [Deepti Chopra, Jacob Perkins, Nitin Hardeniya 2016].



3-rasm. NLTK paketidagi stemming usullari

Lancaster stemming algoritmi deb nomlanuvchi yana bir

stemming algoritmi Lancaster universitetida ishlab chiqilgan. PorterStemmer sinfiga o'xshab, LancasterStemmer sinfi ham **NLTK**da Lancaster stemmingni amalga oshirish uchun ishlatiladi. Biroq ikkita algoritm o'rtasidagi asosiy farq shundaki, Lancaster stemming Porter Stemming bilan solishtirganda turli xil xususiyatlarning ko'proq so'zlaridan foydalanishni o'z ichiga oladi.

```
from nltk.stem import LancasterStemmer
stemmerlan=LancasterStemmer()
print(stemmerlan.stem('removing'))
print(stemmerlan.stem('happies'))
```

```
remov
happy
```

Biz **RegexpStemmer**dan foydalanib NLTKda o'z stemmerimizni ham qurishimiz mumkin. U satrni qabul qilish va moslik topilganda so'zning prefiksi yoki qo'shimchasini olib tashlash orqali ishlaydi. NLTKda RegexpStemmerdan foydalanib, stemming misolini ko'rib chiqamiz:

```
import nltk
from nltk.stem import RegexpStemmer
stemmerregex=RegexpStemmer('ing')
print(stemmerregex.stem('removing'))
print(stemmerregex.stem('happiness'))
print(stemmerregex.stem('pairing'))
```

```
remov
happiness
pair
```

Biz **RegexpStemmer**dan PorterStemmer va LancasterStemmer yordamida o'zakni aniqlash mumkin bo'lmagan hollarda foydalanishimiz mumkin. **SnowballStemmer** ingliz tilidan tashqari **13** tilda stemmingni bajarish uchun ishlatiladi. SnowballStemmer yordamida stemmingni amalga oshirish uchun, birinchi navbatda, stemming bajarilishi kerak bo'lgan *tilni* ko'rsatish kerak. Keyin **stem()** metodidan foydalanib, stemming amalga oshiriladi. SnowballStemmer yordamida NLTKda *ispan* va *fransuz* tillarida stemming jarayoni quyidagicha amalga oshiriladi:

```
import nltk
from nltk.stem import SnowballStemmer
print(" ".join(SnowballStemmer.languages)) # qaysi tillar
qo'llab-quvvatlanishini ko'rish
```

```
spanishstemmer=SnowballStemmer('spanish') # tilni tan-
lash
print(spanishstemmer.stem('comiendo')) # so'z stemmingi
frenchstemmer=SnowballStemmer('french') # tilni tanlash
print(frenchstemmer.stem('manger')) # so'z stemmingi
NLTKda mavjud bo'lgan quyidagi kodni ko'rib chiqamiz, bu
bizga stemmingni amalga oshirish imkonini beradi:
Class StemmerI(object):
    """
    Tokenlardan morfologik affikslarni olib tashlashga yordam
    beradigan interfeys bo'lib, jarayon stemming deb nomlanadi.
    """
    def stem(self, token):
        """
        Tokendan affikslarni olib tashlash va o'zakni qaytarish.
        """
        raise NotImplementedError()
    Bir nechta stemmerlar yordamida stemmingni amalga os-
    hirish uchun ishlatiladigan kodni ko'rib chiqamiz:
    import nltk
    from nltk.stem.porter import PorterStemmer
    from nltk.stem.lancaster import LancasterStemmer
    from nltk.stem import SnowballStemmer

    def obtain_tokens():
        text_file = open("D:/Examples/Examples.txt", "r")
        stem = text_file.read()
        text_file.close()
        tok = nltk.word_tokenize(stem)
        return tok

    def stemming(filtered):
        stem=[]
        for x in filtered:
            stem.append(PorterStemmer().stem(x))
        return stem

    tok=obtain_tokens()
    print("tokens is %s")
    stem_tokens= stemming(tok)
    print(stem_tokens)
```

```
print("After stemming is %s")
res=dict(zip(tok,stem_tokens))
print(res)
```

Lemmatizatsiya. Lemmatizatsiya – soʻzshaklning bosh (leksema koʻrinishini) shaklini aniqlash jarayoni. Lemmatizatsiyadan keyin hosil boʻlgan soʻz muhim ahamiyat kasb etadi. *morphy()* metodi orqali **WordNetLemmatizer**da lemmatizatsiya jarayoni amalga oshiriladi. Kiritilgan soʻz WordNetda topilmasa, oʻzgarishsiz qoladi. *pos* argumentda kiritilgan soʻzning soʻz turkumidagi qismini hosil qilinadi. NLTKda lemmatizatsiya jarayonini koʻrib chiqamiz [36; 38-40]:

```
import nltk
from nltk.stem import WordNetLemmatizer
lemmatizer_output=WordNetLemmatizer()
print(lemmatizer_output.lemmatize('working'))
print(lemmatizer_output.lemmatize('working',pos='v'))
print(lemmatizer_output.lemmatize('works'))
```

WordNetLemmatizer kutubxonasi **WordNet** korpusi deb ataladigan axborot tizimi (ontologiya)dan foydalanib, *morphy()* metodi orqali lemmani hosil qiladi. Agar lemma hosil qilinmasa, soʻz boshlangʻich (asl) shaklda qaytariladi. Masalan, **works** uchun qaytarilgan lemma birlik shakli: **work**. Oʻzbek tilida **kitoblar** soʻzining lemmasi stemi **kitob** boʻladi. Quyidagi dastur kodida stemming va lemmatizatsiya jarayonlari oʻrtasidagi farq koʻrsatadigan:

```
import nltk
from nltk.stem import PorterStemmer
stemmer_output=PorterStemmer()
print(stemmer_output.stem('happiness'))

from nltk.stem import WordNetLemmatizer
lemmatizer_output=WordNetLemmatizer()
print(lemmatizer_output.lemmatize('happiness'))
```

```
happi
happiness
```

Oldingi kodda **happiness** stemming jarayoni natijasida **happi** ga aylandi. Lemmatizatsiya **happiness** soʻzining oʻzagini topa olmaydi. Shuning uchun u **happiness** soʻzini qaytaradi. Oʻzbek tilida stemmatizatsiya jarayoni nisbatan oson kechadi, chunki oʻzbek tilida lemmalar koʻpincha stemga mos keladi. Ammo oʻzbek tilida ham

fleksiya holatlari uchurab turadi. Masalan: *yutug'ı, so'rogim* kabi so'zlarda o'zak *yutug', so'rog'* emas, balki *yutuq, so'roq* shaklida bo'ladi. Bunday holatlarda stemni topishning yuqoridagi kabi usulidan foydalaniladi.

Polyglot – tokenlardan morfemalarni aniqlash uchun ishlatiladigan *morfessor modellari* deb ataladigan modellarni taqdim etish uchun ishlatiladigan dasturiy ta'minot [Al-Rfou, Perozzi, Skiena 2013; Thanaki 2017]. **Morpho** loyihasining maqsadi ma'lumotlarga qayta ishlash uchun boshqarilmaydigan jarayonlarni hosil qilishdan iborat. Morfo loyihasi asosida sintaksisning eng kichik birligi bo'lgan morfemalar yaratiladi. Tabiiy tilni qayta ishlashda morfemalar muhim rol o'ynaydi. Morfemalardan tilni avtomatik tanib olish va yaratishda foydalaniladi. *Polyglot*ning lug'atlari yordamida turli tillarning 50000 ta tokenlarida morfessor modellari ishlab chiqilgan.

Morfologik analizator

Morfologik tahlil jarayonida tokenlarning qo'shimchalarga asoslangan ma'nolarini hisobga olgan holda grammatik ma'lumot olish amalga oshiriladi. Morfologik tahlil uchta usulda amalga oshirilishi mumkin:

- 1) *morfemaga asoslangan morfologiya;*
- 2) *leksema asosidagi morfologiya;*
- 3) **ordga** asoslangan morfologiya.

Morfologik analizator muayyan tokenning morfologik tarkibini tahlil qilish uchun mas'ul bo'lgan dastur sifatida izohlanadi. Morfologik analizator berilgan tokenni tahlil qiladi va turkum, turli grammatik ma'nolar kabi ma'lumotlarni shakllantiradi. Berilgan bo'sh joy bo'lmagan tokenda morfologik tahlilni amalga oshirish uchun **pyEnchant** lug'ati qo'llaniladi. So'zning turlumini aniqlash uchun qator qoidalar guruhi talab etiladi. Quyidagi qoidalar asosida so'z turkumini aniqlashimiz mumkin:

1. **Morfologik qoidalar.** Qo'shimchalar haqidagi ma'lumotlar so'z turkumini aniqlashga yordam beradi. Masalan, ingliz tilida **-ness** va **-ment** qo'shimchalari otlar bilan birikadi. Demak, ingliz tilida so'z oxorodagi yasovchi qo'shimchaga qarab uning turkumini aniqlash mumkin. Ammo o'zbek tilida bu usul bilan so'zning turkumini aniqlash imkoni yo'q. Shu sababli o'zbek tili morfoleksikoni ishlab chiqilishi kerak.

2. **Sintaktik qoidalar.** Kontekstli ma'lumotlar so'z turkumini

aniqlashga yordam beradi. Masalan, agar ot turkumiga tegishli bo'lgan so'zni topsak, unda sintaktik qoidalar *sifatning otdan oldin* yoki *ot turkumidan keyin kelishini* aniqlash uchun foydali bo'ladi.

3. **Semantik qoidalar.** Semantik qoidalar so'z turkumini aniqlashda ahamiyatli. Masalan, so'zning joy nomini ifodalashi aniqlansa, uning ot so'z turkimiga mansub ekanligi to'g'risida xulosa qilish mumkin.

4. **Ochiq sinf (guruh).** Ushbu guruhdagi so'zlar ro'yxatiga har kuni yangi so'z qo'shilganda ularning soni ortib boraveradi. Ochiq sinfdagi so'zlar odatda ot so'z turkimiga mansub so'zlardir. Predloglar, asosan, yopiq sinfga mansub bo'ladi. Masalan, *FIO* ro'yxatida cheksiz miqdordagi so'zlar bo'lishi mumkin.

5. **So'z turkumlarini aniqlash (POS, Part of Speech):** So'z turkumlari teglari to'plami morfologik xususiyatni aniqlashga yordam beradigan ma'lumotlarni o'zida jamlaydi.

Masalan, **o'ynaydi** so'zi *uchinchi shaxs birlikdagi ot so'z turkumiga* mansub so'z bilan birga keladi.

6. **Omorfi:** Omorfi paketi GNU GPL 3-versiyasi tomonidan litsenziyalangan bo'lib, *tilni modellashtirish, morfologik tahlil, qoidalariga asoslangan mashina tarjimasi, ma'lumotni qidirish, statistik mashina tarjimasi, morfologik segmentatsiya, ontologiyalar, imloni tekshirish va tuzatish* kabi ko'plab vazifalarni bajarish uchun ishlatiladi.

Morfologik generator

Morfologik generator – morfologik generatsiya vazifasini bajaradigan dastur bo'lib, uni morfologik tahlilning qarama-qarshi vazifasi deb hisoblash mumkin. Bu yerda so'zning **son, turkum, o'zak** va boshqa ma'lumotlariga ko'ra tavsifi berilgan bo'lsa, **asl so'z** hosil qilinadi. Masalan, **o'zak=bormoq, gap bo'lagi = kesim, zamon=hozirgi** va **uchinchi shaxs birlik predmeti** bilan birga kelgan bo'lsa, morfologik generator uning **bormoqda** shaklini hosil qiladi. Morfologik tahlil va generatsiyani amalga oshiradigan ko'plab Pythonga asoslangan dasturiy ta'minotlar mavjud:

1. **ParaMorfo:** U ispan va guarani tilidagi *ot, sifat va fe'llarni* morfologik hosil qilish va tahlil qilish uchun ishlatiladi.

2. **HornMorpho:** Oromo va amhar tillaridagi *ot va fe'llarni*, shuningdek, *tigrinya fe'llarini* morfologik hosil qilish va tahlil qilish uchun ishlatiladi.

3. **AntiMorfo:** Kechua tilidagi *sifatlar, fe'llar va otlarni*, shuningdek, ispan fe'llarini morfologik yaratish va tahlil qilish uchun ish-

latiladi.

4. **MorfoMelayu:** Malay tilidagi so'zlarni morfologik tahlil qilish uchun ishlatiladi.

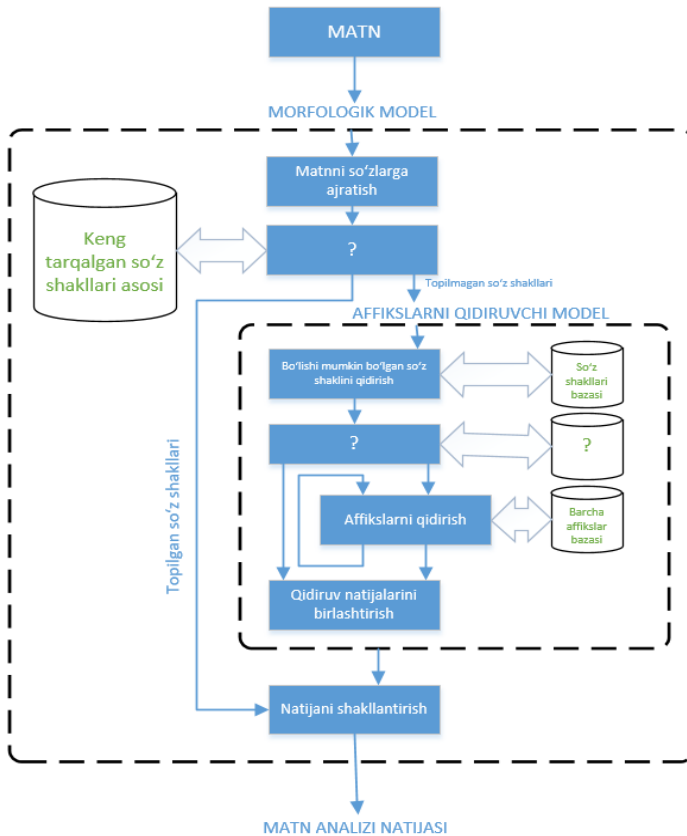
5. **Morph** ingliz tili uchun morfologik generator va analizatori.

6. **Morphy** nemis tili uchun morfologik generator, analizator va POS teggeridir.

7. **Morphisto** nemis tili uchun morfologik generator va analizatoridir.

8. **Morfette** ispan va frantsuz tillarini *nazorat ostida o'rganishni (fleksion morfologiya)* amalga oshiradi.

Yuqoridagi fikr-mulohazalardan matn analizatori dasturiy ta'minotining arxitekturasini (4-rasm)ni quyidagicha shakllantirish mumkin.



4-rasm. Matn analizatori dasturiy ta'minoti arxitekturasini

Xulosa

Kompyuter lingvistikasining ko'plab tarmoqlari mavjud. Matnga ishlov berish (tahlil qilish) yoki generatsiya qilish uchun beril-

gan matn ustida bir qator lingvistik amallarni bajarishi lozim. Ushbu maqolada **stemming, lemmatizatsiya va morfologik tahlil hamda generatsiya** amallarini NLTK paketi vositalari yordamida amalga oshirish ko'rib chiqildi. Shuningdek, qidiruv tizimlari va ularni amalga oshirish usullari muhokama qilindi.

Adabiyotlar

- Öztürel, A., Kayadelen, T., & Demirşahin, I. 2019. A syntactically expressive morphological analyzer for turkish. *FSMNLP 2019 - 14th International Conference on Finite-State Methods and Natural Language Processing, Proceedings*. <https://doi.org/10.18653/v1/w19-3110>
- Jabbar, A., Iqbal, S., Akhunzada, A., & Abbas, Q. 2018. An improved Urdu stemming algorithm for text mining based on multi-step hybrid approach. *Journal of Experimental and Theoretical Artificial Intelligence*, 30(5). <https://doi.org/10.1080/0952813X.2018.1467495>
- Szabó, G., & Kovács, L. (2018). Benchmarking morphological analyzers for the Hungarian language. *Annales Mathematicae et Informaticae*, 49. <https://doi.org/10.33039/ami.2018.05.001>
- Daniel S. Jurafsky and James H. Martin. Speech and Language Processing. Contributing writers: Andrew Kehler, Keith Vander Linden, Nigel Ward 2000y. Prentice Hall, Englewood Cliffs, New Jersey 07632. pages: 950.
- Mohri M.A. Finite-state transducers in language and speech processing. *Computational Linguistics*. – С. 269-312.
- Ахо Альфред В., Лам Моника С, Сети Рави, Ульман Джеффри Д. 2008. Компиляторы: принципы, технологии и инструментарий, 2-е изд.: Пер. с англ. – М.: ООО “И.Д. Вильямс”, 1184 с.
- Толдова С.Ю., Бонч-Осмоловская А.А. Автоматический морфологический анализ // Фонд знаний «Ломоносов». М., 2011. www.lomonosov-fund.ru/enc/ru/encyclopedia:0127430
- Садыков Т., Кочконбаева Б. 2018. Об оптимизации алгоритма морфологического анализа // Шестая Международная конференция по компьютерной обработке тюркских языков «Turklang-2018». (Труды конференции) – Ташкент, 320 с.
- Rodolfe, D. 2008. Computational Linguistic Text Processing: Lexicon, Grammar, Parsing and Anaphora Resolution. NovaSciencePublishers, Inc. – NewYork, – P. 4-5.
- Ермаков А. Морфологический анализатор - основа поисковых систем // <https://www.kv.by/archive/index2004154301.htm>
- Ножов И.М. 2003. Морфологическая и синтаксическая обработка текста (модели и программы): диссертация канд. наук. Москва, 190 с.
- Дыбо А.В., Шеймович А.В. 2014. Автоматический морфологический

- анализ для корпусов хакасского и древнетюркского языков / Научное обозрение саяно-алтая рецензируемый научный журнал Номер 2(08), (Периодичность – 2 раза в год. серия: Филология.Выпуск 2). С. 9-31.
- Сулейманов Д.Ш., Гатиатуллин А.Р. 1996. Модель татарской аффиксальной морфемы и ее реализация // Серия: Интеллект. Язык. Компьютер. Вып.4. Казань: Фэн. 113-127.
- Сулейманов Д.Ш., Гильмуллин Р.А., Гатауллин Р.Р. 2017. Морфологический анализатор татарского языка на основе двухуровневой модели морфологии / Пятая Международная конференция по компьютерной обработке тюркских языков «TurkLang 2017». – Труды конференции. В 2-х томах. Т 2. – Казань: Издательство Академии наук Республики Татарстан, 327 с.;
- Желтов П.В. 2017. Разработка морфологического анализатора чувашского языка // Пятая Международная конференция по компьютерной обработке тюркских языков «TurkLang 2017». – Труды конференции. В 2-х томах. Т 2. – Казань: Издательство Академии наук Республики Татарстан, 327 с.
- Исраилова Н.А., Бакасова П.С. 2017. Морфологический анализатор кыргызского языка // Пятая Международная конференция по компьютерной обработке тюркских языков «TurkLang 2017». – Труды конференции. В 2-х томах. Т 2. – Казань: Издательство Академии наук Республики Татарстан, 327 с.
- Леонтьев Н.А. 2018. Морфологический анализатор якутского языка // Шестая Международная конференция по компьютерной обработке тюркских языков «TurkLang-2018». (Труды конференции) – Ташкент, 320 с. – С. 276-279.
- Куканова В.В., Каджиев А.Ю. 2014. Алгоритм работы морфологического парсера калмыцкого языка // В сборнике: Писменото наследство и информационните технологии. E\Manuscript-2014 Материали от V международна научной конференции. С. 116-119.
- Орхун М. 2018. Computational analysis of uzbek nouns / Шестая Международная конференция по компьютерной обработке тюркских языков «TurkLang-2018». (Труды конференции) Ташкент, 320 с.
- Турсунов А. 2018. Вопросы словообразования в формальных моделях тюркских языков (на примере узбекского языка) // Шестая Международная конференция по компьютерной обработке тюркских языков «TurkLang-2018». Труды конференции) Ташкент, – 320 с.
- Болтаев Т.Б., Ибрагимов С.И. 1992. О проекте программной системы морфологического анализа узбекского языка // <http://buxdu.uz/index.php/uz/>; Болтаев Т.Б., Кузьминов Т.В.,

- Поттосин И.В. О структурном конструировании программ и инструментах его поддержки // Среда программирования: методы и инструменты. Новосибирск, С22-37.
- Gökçe, F. 2014. Altay dillerinde ortak bir gramatikal morphem üzerine. *Turkbilig*, 27.
- Ruppenhofer, J., Steiner, P., & Wiegand, M. 2017. Evaluating the morphological compositionality of polarity. *International Conference Recent Advances in Natural Language Processing, RANLP, 2017-September*. <https://doi.org/10.26615/978-954-452-049-6-081>
- Minegishi, M. 2011. Description of Thai as an isolating language. *Social Science Information*, 50(1). <https://doi.org/10.1177/0539018410389107>
- Pan, Y., Li, X., Yang, Y., & Dong, R. 2020. Multi-source neural model for machine translation of agglutinative language. *Future Internet*, 12(6). <https://doi.org/10.3390/FI12060096>
- Hakkani-Tür, D. Z., Oflazer, K., & Tur, G. 2002. Statistical morphological disambiguation for agglutinative languages. *Language Resources and Evaluation*, 36(4). <https://doi.org/10.3115/990820.990862>
- Kelly, B., Wigglesworth, G., Nordlinger, R., & Blythe, J. 2014. The acquisition of polysynthetic languages. *Linguistics and Language Compass*, 8(2). <https://doi.org/10.1111/lnc3.12062>
- Кузнецов П.С., 2010. Морфологическая классификация языков, Материалы к курсам языкознания
- Bareket, D., & Tsarfaty, R. 2021. Neural modeling for named entities and morphology (Nemo2). *Transactions of the Association for Computational Linguistics*, 9. https://doi.org/10.1162/tac-a_00404
- Ledeneva, Y., & Sidorov, G. 2010. Recent advances in computational linguistics. *Informatica (Ljubljana)*, 34(1).
- Pirkola, A. 2001. Morphological typology of languages for IR. *Journal of Documentation*, 57(3). <https://doi.org/10.1108/EUM00000000007085>
- P. Jayan, J., R. R., & Rajendran, Dr. S. 2011. Morphological Analyser and Morphological Generator for Malayalam Tamil Machine Translation. *International Journal of Computer Applications*, 13 (8). <https://doi.org/10.5120/1802-2440>
- Aslan, O., Gunal, S., & Dincer, B. T. 2018. A computational morphological lexicon for Turkish: TrLex. *Lingua*, 206. <https://doi.org/10.1016/j.lingua.2018.01.003>
- Rifano, E. J., Fauzan, Abd. C., Makhi, A., Nadya, E., Nasikin, Z., & Putra, F. N. (2020). Text Summarization Menggunakan Library Natural Language Toolkit (NLTK) Berbasis Pemrograman Python. *IL-KOMNIKA: Journal of Computer Science and Applied Informatics*, 2(1). <https://doi.org/10.28926/ilkomnika.v2i1.32>
- Yayla, M., Diyar Demirkol, M., & Alqaraleh, S. (2021). CNN vs. LSTM for Turkish text classification. *2021 International Conference on INnovations in Intelligent SysTems and Applications, INI-*

- STA 2021 - Proceedings. <https://doi.org/10.1109/INIS-TA52262.2021.9548407>
- Deepti Chopra, Jacob Perkins, Nitin Hardeniya. Natural Language Processing: Python and NLTK. (2016). Publisher(s): Packt Publishing. ISBN: 9781787285101
- Orhun M., Tantug C., Adali E. Rule Based Analysis of the Uyghur Nouns, International Journal of Asian Lang. Proc.,19(1), 33-44, 2009.
- Khyani, D., 2021. An Interpretation of Lemmatization and Stemming in Natural Language Processing. *Journal of University of Shanghai for Science and Technology*, 22(10).
- Balakrishnan, V., & Ethel, L.-Y. (2014). Stemming and Lemmatization: A Comparison of Retrieval Performances. *Lecture Notes on Software Engineering*, 2(3). <https://doi.org/10.7763/lnse.2014.v2.134>
- Balakrishnan, V., & Ethel, L.-Y. (2014). Stemming and Lemmatization: A Comparison of Retrieval Performances. *Lecture Notes on Software Engineering*, 2(3). <https://doi.org/10.7763/lnse.2014.v2.134>
- Al-Rfou, R., Perozzi, B., & Skiena, S. (2013). Polyglot: Distributed word representations for multilingual NLP. *CoNLL 2013 - 17th Conference on Computational Natural Language Learning, Proceedings*.
- Thanaki, J. 2017. Python Natural Language Processing. In *Packt Publishing Ltd*.
- Çağrı Çöltekin. 2014. A Set of Open Source Tools for Turkish Natural Language Processing In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14) Ed. by N. Calzolari et al. 1079-1086.
- Kessikbayeva, G. and I. Cicekli, 1994. "Rule based morphological analyzer of Kazakh language," in Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM. Baltimore, Maryland: ACL, June 2014, pp. 46-54.;
- Kemal Oflazer. Two-level Description of Turkish Morphology. Literary and Linguistic Computing, – Vol. 9, No 2.
- Tantug C., Adali E., 2006. Oflazer K. Computer Analysis of the Turkmen Language Morphology, 5th International Conference on NLP (FinTAL 2006), Turku, Finland, 186-193.

Methods of creating a morphological analysis

Botir Elov¹

Shaxlo Hamroeva²

Dilrabo Elova³

Abstract

The process of morphological analysis is an important component of natural language processing systems, such as spelling correctors, parsers, machine translation systems, and electronic dictionaries. This article describes the steps of a text analyzer, a morphological analyzer, and how to create a morphological generator. Here's how to use the Python NLTK package tools to create a morphological analyzer, and examples of program code. The structure and architecture of the morphological analyzer are also presented on the basis of the stages of the morphological analysis process (inflection, derivation, detection of affixids, compound forms, cliticization).

Key words: *Natural language processing, NLP, Python, morphological analyzer, token, lemmatization, stemming, morphological generator, search engine, stemming algorithm, PorterStemmer.*

References

- Öztürel, A., Kayadelen, T., & Demirşahin, I. 2019. A syntactically expressive morphological analyzer for turkish. FSMNLP 2019 - 14th International Conference on Finite-State Methods and Natural Language Processing, Proceedings. <https://doi.org/10.18653/v1/w19-3110>
- Jabbar, A., Iqbal, S., Akhunzada, A., & Abbas, Q. 2018. An improved Urdu stemming algorithm for text mining based on multi-step hybrid approach. Journal of Experimental and Theoretical Artificial Intelligence, 30(5). <https://doi.org/10.1080/095281>

¹Elov Botir Boltaevich – Tashkent State University of Uzbek Language and Literature named after Alisher Navoi - Head of the Department of Computer Linguistics and Digital Technologies, PhD in Engineering

E-mail: elov@navoiy-uni.uz

ORCID: 0000-0001-5032-6648

²Hamroeva Shaxlo Mirdjonovna – Tashkent State University of Uzbek Language and Literature named after Alisher Navoi, Department of Applied Philology, Doctor of Philology (DSc)

E-mail: shaxlo.xamrayeva@navoiy-uni.uz

ORCID: 0000-0002-5429-4708

³Elova Dilrabo Qudratilloevna - Tashkent State University of Uzbek Language and Literature named after Alisher Navoi, teacher of the Department of Applied Philology

E-pochta: elovadilrabo@navoiy-uni.uz

ORCID: 0000-0002-2329-1811

For citation: Elov, B.B., Hamroeva, Sh. M., Elova D.Q. 2022. "Methods of creating a morphological analyzer". *Uzbekistan: language and culture. Applied Philology* 1 (5): 88-108.

3X.2018.1467495

- Szabó, G., & Kovács, L. (2018). Benchmarking morphological analyzers for the Hungarian language. *Annales Mathematicae et Informaticae*, 49. <https://doi.org/10.33039/ami.2018.05.001>
- Daniel S. Jurafsky and James H. Martin. *Speech and Language Processing*. Contributing writers: Andrew Kehler, Keith Vander Linden, Nigel Ward 2000y. Prentice Hall, Englewood Cliffs, New Jersey 07632. pages: 950.
- Mohri M.A. Finite-state transducers in language and speech processing. *Computational Linguistics*. – S. 269-312.
- Axo Alfred V., Lam Monika S, Seti Ravi, Ul'man Djeffri D. 2008. *Kompilyatori: printsiipi, texnologii i instrumentariy*, 2-e izd.: Per. s angl. – M.: OOO "I.D. Vilyamc", 1184 s.
- Toldova S.Yu., Bonch-Osmolovskaya A.A. *Avtomaticheskiy morfologicheskii analiz* // Fond znaniy «Lomonosov». M., 2011. www.lomonosov-fund.ru/enc/ru/encyclopedia:0127430
- Sadikov T., Kochkonbaeva B. 2018. *Ob optimizatsii algoritma morfologicheskogo analiza* // *Shestaya Mejdunarodnaya konferentsiya po kompyuternoy obrabotke tyurkskix yazikov «Turklang-2018»*. (Trudi konferentsii) – Tashkent, 320 s.
- Rodolfe, D. 2008. *Computational Linguistic Text Processing: Lexicon, Grammar, Parsing and Anaphora Resolution*. NovaSciencePublishers, Inc. – NewYork, – P. 4-5.
- Yermakov A. *Morfologicheskii analizator - osnova poiskovix sistem* // <https://www.kv.by/archive/index2004154301.htm>
- Nojov I.M. 2003. *Morfologicheskaya i sintaksicheskaya obrabotka teksta (modeli i programmi): dissertatsiya kand. nauk*. Moskva, 190 s.
- Dibo A.V., Sheymovich A.V. 2014. *Avtomaticheskiy morfologicheskii analiz dlya korpusov xakasskogo i drevnetyurkskogo yazikov / Nauchnoe obozrenie sayano-altaya retsenziruemyy nauchniy jurnal Nomer 2(08), (Periodichnost – 2 raza v god. seriya: Filologiya. Vipusk 2). S. 9-31.*
- Suleymanov D.Sh., Gatiatullin A.R. 1996. *Model tatarskoy affiksallyy morfemy i yee realizatsiya* // *Seriya: Intellect. Yazik. Komp'yuter. Vip.4. Kazani: Fen. 113-127.*
- Suleymanov D.Sh., Gilimullin R.A., Gataullin R.R. 2017. *Morfologicheskii analizator tatarskogo yazyka na osnove dvuxurovnevoy modeli morfologii* / *Pyataya Mejdunarodnaya konferentsiya po kompyuternoy obrabotke tyurkskix yazikov «TurkLang 2017»*. – Trudi konferentsii. V 2-x tomax. T 2. – Kazan: Izdatelstvo Akademii nauk Respubliki Tatarstan, 327 s.;
- Jeltov P.V. 2017. *Razrabotka morfologicheskogo analizatora chuvashskogo yazyka* // *Pyataya Mejdunarodnaya konferentsiya po kompyuternoy obrabotke tyurkskix yazikov «TurkLang 2017»*. – Trudi konferentsii. V 2-x tomax. T 2. – Kazan: Izdatelstvo Akademii nauk Respubliki Tatarstan, 327 s.
- Israilova N.A., Bakasova P.S. 2017. *Morfologicheskii analizator kirgizskogo yazyka* // *Pyataya Mejdunarodnaya konferentsiya po kompyuternoy obrabotke tyurkskix yazikov «TurkLang 2017»*. – Trudi konferentsii. V 2-x tomax. T 2. – Kazani: Izdatelstvo Akademii nauk Respubliki Tatarstan, 327 s.

- Leont'ev N.A. 2018. Morfologicheskii analizator yakutskogo yazika // Shestaya Mejdunarodnaya konferentsiya po kompyuternoy obrabotke tyurkskix yazikov «TurkLang-2018». (Trudi konferentsii) – Tashkent, 320 s. – S. 276-279.
- Kukanova V.V., Kadjiev A.Yu. 2014. Algoritm raboti morfologicheskogo parsera kalmitskogo yazыka // V sbornike: Pismenoto nasledstvo i informatsionnite texnologii. El'Manuscript-2014 Materiali ot V mejdunarodna nauchnoy konferentsii. S. 116-119.
- Orxun M. 2018. Computational analysis of uzbek nouns / Shestaya Mejdunarodnaya konferentsiya po kompyuternoy obrabotke tyurkskix yazikov «TurkLang-2018». (Trudi konferentsii) Tashkent, 320 s.
- Tursunov A. 2018. Voprosi slovoobrazovaniya v formalnix modelyax tyurkskix yazikov (na primere uzbekskogo yazika) // Shestaya Mejdunarodnaya konferentsiya po kompyuternoy obrabotke tyurkskix yazikov «TurkLang-2018». Trudi konferentsii) Tashkent, – 320 s.
- Boltaev T.B., Ibragimov S.I. 1992. O proekte programmnoy sistemy morfologicheskogo analiza uzbekskogo yazika // <http://buxdu.uz/index.php/uz/>; Boltaev T.B., Kuzminov T.V., Pottosin I.V. O strukturnom konstruirovaniye programm i instrumentax yego podderjki // Sreda programmirovaniya: metodi i instrumenti. Novosibirsk, S22-37.
- Gökçe, F. 2014. Altay dillerinde ortak bir gramatikal morfem üzerine. *Turkbilig*, 27.
- Ruppenhofer, J., Steiner, P., & Wiegand, M. 2017. Evaluating the morphological compositionality of polarity. *International Conference Recent Advances in Natural Language Processing, RANLP*, 2017-September. <https://doi.org/10.26615/978-954-452-049-6-081>
- Minegishi, M. 2011. Description of thai as an isolating language. *Social Science Information*, 50(1). <https://doi.org/10.1177/0539018410389107>
- Pan, Y., Li, X., Yang, Y., & Dong, R. 2020. Multi-source neural model for machine translation of agglutinative language. *Future Internet*, 12(6). <https://doi.org/10.3390/FI12060096>
- Hakkani-Tür, D.Z., Oflazer, K., & Tur, G. 2002. Statistical morphological disambiguation for agglutinative languages. *Language Resources and Evaluation*, 36(4). <https://doi.org/10.3115/990820.990862>
- Kelly, B., Wigglesworth, G., Nordlinger, R., & Blythe, J. 2014. The acquisition of polysynthetic languages. *Linguistics and Language Compass*, 8(2). <https://doi.org/10.1111/lnc3.12062>
- Kuznetsov P.S., 2010. Morfologicheskaya klassifikatsiya yazikov, Materiali k kursam yazikoznaniya
- Bareket, D., & Tsarfaty, R. 2021. Neural modeling for named entities and morphology (Nemo2). *Transactions of the Association for Computational Linguistics*, 9. https://doi.org/10.1162/tacl_a_00404
- Ledeneva, Y., & Sidorov, G. 2010. Recent advances in computational linguistics. *Informatica (Ljubljana)*, 34(1).
- Pirkola, A. 2001. Morphological typology of languages for IR. *Journal of Documentation*, 57(3). <https://doi.org/10.1108/>

EUM0000000007085

- P.Jayan, J., R. R. R., & Rajendran, Dr. S. 2011. Morphological Analyser and Morphological Generator for Malayalam Tamil Machine Translation. *International Journal of Computer Applications*, 13(8). <https://doi.org/10.5120/1802-2440>
- Aslan, O., Gunal, S., & Dincer, B. T. 2018. A computational morphological lexicon for Turkish: TrLex. *Lingua*, 206. <https://doi.org/10.1016/j.lingua.2018.01.003>
- Rifano, E. J., Fauzan, Abd. C., Makhi, A., Nadya, E., Nasikin, Z., & Putra, F. N. (2020). Text Summarization Menggunakan Library Natural Language Toolkit (NLTK) Berbasis Pemrograman Python. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 2(1). <https://doi.org/10.28926/ilkomnika.v2i1.32>
- Yayla, M., Diyar Demirkol, M., & Alqaraleh, S. (2021). CNN vs. LSTM for Turkish text classification. 2021 International Conference on INnovations in Intelligent SysTems and Applications, INISTA 2021 - Proceedings. <https://doi.org/10.1109/INISTA52262.2021.9548407>
- Deepti Chopra, Jacob Perkins, Nitin Hardeniya. *Natural Language Processing: Python and NLTK*. (2016). Publisher(s): Packt Publishing. ISBN: 9781787285101
- Orhun M., Tantug C., Adali E. Rule Based Analysis of the Uyghur Nouns, *International Journal of Asian Lang. Proc.*,19(1), 33-44, 2009.
- Khyani, D., 2021. An Interpretation of Lemmatization and Stemming in Natural Language Processing. *Journal of University of Shanghai for Science and Technology*, 22(10).
- Balakrishnan, V., & Ethel, L.-Y. (2014). Stemming and Lemmatization: A Comparison of Retrieval Performances. *Lecture Notes on Software Engineering*, 2(3). <https://doi.org/10.7763/lnse.2014.v2.134>
- Balakrishnan, V., & Ethel, L.-Y. (2014). Stemming and Lemmatization: A Comparison of Retrieval Performances. *Lecture Notes on Software Engineering*, 2(3). <https://doi.org/10.7763/lnse.2014.v2.134>
- Al-Rfou, R., Perozzi, B., & Skiena, S. (2013). Polyglot: Distributed word representations for multilingual NLP. *CoNLL 2013 - 17th Conference on Computational Natural Language Learning, Proceedings*.
- Thanaki, J. 2017. *Python Natural Language Processing*. In Packt Publishing Ltd.
- Çağrı Çöltekin. 2014. A Set of Open Source Tools for Turkish Natural Language Processing In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC14)* Ed. by N. Calzolari et al. 1079-1086.
- Kessikbayeva, G. and I. Cicekli, 1994. "Rule based morphological analyzer of Kazakh language," in *Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM*. Baltimore, Maryland: ACL, June 2014, pp. 46-54.;
- Kemal Oflazer. Two-level Description of Turkish Morphology. *Literary and Linguistic Computing*, – Vol. 9, No 2.
- Tantug C., Adali E., 2006. Oflazer K. Computer Analysis of the Turkmen Language Morphology, 5th International Conference on NLP (FinTAL 2006), Turku, Finland, 186-193.