УДК: 81'322

Касиева А. А., ф. и. к., доц. aida.kasieva@manas.edu.kg
Абитова А. А., 2-курс. студенти, 1901.10010@manas.edu.kg
«Манас» КТУ

ТАБИГЫЙ ТИЛ ИШТЕТҮҮДӨ (ТТИ) ЛЕММАТИЗАЦИЯ ПРОЦЕССИНИН ОРДУ

Бул макалабыз табигый тилди иштетүүдө лемматизация процессинин орду жана кыргыз тилиндеги сөздөргө кантип лемматизация жүргүзүү керектиги тууралуу маалымат сунуштайт. Макалада башка тилдердеги сөздөр компьютердик программалоонун жардамы менен кантип талданса, кыргыз тилиндеги сөздөргө да дал ошондой талдоо жүргүзүүгө мүмкүнбү деген суроого жооп изделет. Кыргыз тили агглютинативдүү, башкача айтканда, улама, уңгуга мүчө уланып жаңы сөздөр жасалуучу касиетке ээ тил болгондуктан, аны лемматизациялоодо бир топ кыйынчылыктар жаралат. Ошондуктан алгач англис тилинин лемматизациясы жана аны аткаруудагы пайдаланылган Python пакеттери жөнүндө сөз болот. Ошондой эле кыргыз тили лексикалык составы, синтаксистик түзүлүшү жана морфологиялык каражаттары жагынан түрк тилине жакын болгондуктан, түрк лемматизациясы жөнүндө айтылып, эки тилдеги сөздөрдү лемматизациялоодо кетирилген каталардын окшоштугуна, кыйынчылыктардын бирдейлигине маани берилет.

Өзөктүү сөздөр: *Табигый тил иштетүү, лемматизация, лемма, стемминг, стем, Python пакети, Spacy, ачык булак китепкана, синтаксистик талдоо.*

Касиева А. А., к. ф. н., доцент, aida.kasieva@manas.edu.kg
Абитова А. А., студент 2го курса, 1901.10010@manas.edu.kg
КТУ «Манас»

РОЛЬ ЛЕММАТИЗАЦИИ В ОБРАБОТКЕ ЕСТЕСТВЕННОГО ЯЗЫКА (НЛП)

В этой статье рассматривается процесс лемматизации при обработке естественного языка (NLP) и способы лемматизации слов в кыргызском языке. В статье делается попытка ответить на вопрос, могут ли слова кыргызского языка обрабатываться аналогичным образом с помощью компьютерного программирования. Кыргызский язык агглютинативен, то есть, он может постоянно производить новые словоформы, добавляя аффиксы к основе слова. Следовательно, это вызывает определенные трудности при проведении процесса лемматизации. Поэтому в этом исследовании сначала мы рассматриваем лемматизацию английского языка с использованием пакетов Python. Затем мы рассматриваем лемматизацию в турецком языке в связи с его близостью к кыргызскому языку с точки зрения синтаксической структуры и сходства морфологических признаков. Таким образом, выявляются схожесть ошибок и трудности их лемматизации.

Ключевые слова: *НЛП, лемматизация, лемма, стемминг, основа, пакеты Python, Spacy, синтаксический анализ, библиотека с открытым доступом, синтаксический анализ.*

Kasieva A. candidate of Philology Sciences, Associate Prof
aida.kasieva@manas.edu.kg
A.Abitova 2nd grade student 1901.10010@manas.edu.kg
Kyrgyz-Turkish 'Manas' University

THE ROLE OF LEMMATIZATION IN NATURAL LANGUAGE PROCESSING (NLP)

This article deals with the process of lemmatization in Natural Language Processing (NLP) and how to lemmatize words in the Kyrgyz language. The article seeks to answer the question of whether words in the Kyrgyz language can be processed in the analogical way by computer programming. The Kyrgyz language is agglutinative, that is, it can continuously produce new word forms by adding affixes to the stem. Consequently, this causes certain challenges in carrying out lemmatization process. Hence, in this study, initially we consider lemmatization of the English language, using Python packages. Then we consider lemmatization in the Turkish language due to its closeness with the Kyrgyz language in terms of syntactic structure and similarity of morphological means. Thus, the similarity of errors and difficulties in their lemmatization are identified.

Key words: *NLP, lemmatization, lemma, stemming, stem, Python packages, Spacy, open-source library, syntactic analysis.*

Табигый тил иштетүү (ТТИ) деген эмне? ТТИ (NLP) – лингвистиканын, информатиканын жана жасалма интеллектинин компьютерлер менен адамдын өз ара байланышын жөнгө салган бөлүм. Тагыраак айтканда, табигый тилдеги маалыматтарды иштетүү жана талдоо үчүн компьютерлерди кантип программалоо керек деген суроого жооп берет. Ал эми лемматизация сөздү мүчөлөрүнөн ажыратып, анын леммасын, башкача айтканда, сөздүн уңгусун/негизин бөлүп чыгарат. Ал табигый тилди иштетүүдө ар кайсы тилде ар башка метод, ар түрдүү Python пакеттери аркылуу аткарылат. Бул анализ адам тарабынан программаланган же үйрөтүлгөн компьютердин жардамы аркылуу ишке ашырылат.

Лемматизация табигый тилди иштетүүдө сөздүн леммасын анын маанисине жана контекстке жараша тапкан алгоритмдик процесс болуп саналат. Ал сөздүн морфологиялык анализин билдирип, сөз өзгөртүүчү мүчөлөрдү жоюуга багытталат. Андагы сөздүн негизи *лемма* деп аталып, сөздү уңгуга же сөздүктөгү формасына келтирүүгө жардам берет. М.: Баланын машиналарынын өңү ар түстө = Бала машина өң ар түс. [https://www.https://www.machinelearninghlus.com/nlp/lemmatization-examples-python/ machinelearninghlus.com/nlp/lemmatization-examples-python/].

Лемматизация процессинин стеммингден айырмасы

Лемматизация сөздү уңгуга алып келүү процесси жана анын стеммингден айырмасы - сөздү контекстке жараша маани берген формасына келтирүүдө болуп саналат. Ал эми стемминг сөздүн аягына уланган мүчөлөрдү гана жойгондуктан, анда маанилик жактан жана орфографиялык каталар кетиши мүмкүн. М.: Лемматизация "caring" ден "care" деп уңгуну туура аныктайт. Стемминг болсо "ing" мүчөсүн гана кыскартып, "car" негизин бөлүп көрсөтөт.

"Caring"- лемматизация -"care"

"Caring"- стемминг -"car"

Change, changing, changes, changed, changer} chang - стемминг.

Change, changing, changes, changed, changer} change - лемматизация.

Лемматизация айрым Python пакеттери аркылуу ишке ашырылат. Айрым учурда кээ бир окшош сөздөрдүн ар башка леммасы болгону кездешет. Ошондуктан алардын сүйлөмдө колдонулушуна жараша сөз түркүмү (POS) аныкталып, туура лемма берилет. Мындай талдоонун ишке ашырылышы төмөнкү бөлүмдөр аркылуу жүргүзүлөт жана лемматизацияны ишке ашырууда төмөнкү Python пакеттери колдонулат:

1. Wordnet POS (Parts of speech) tag лемматизатор

Wordnet англис тили үчүн кеңири жана эркин колдонулган жалпыга жеткиликтүү лексикалык база болуп саналат. Ал сөздөрдүн ортосуна структуралаштырылган семантикалык байланыштарды орнотууга багытталган. Бул пакет лемматизация мүмкүнчүлүктөрүн сунуштаган эң алгачкы жана көп колдонулган лемматизатор катары белгилүү. Ал эми андагы NLTK (Natural language toolkit) (табигый тил куралдары) ага өз ара байланыш (интерфейс) сунуштайт. Бирок аны колдонуу үчүн, аны алгач жүктөп алуу зарыл. Жогоруда аталган курал жана тиркемени колдонуп, биз төмөнкү сүйлөмдү компьютерге берилген код-программа аркылуу талдап көрөлү:

[https://www.machinelearninghlus.com/nlp/lemmatization-examples-python/]

#> [‘Чаар’, ‘жарганаттар’, өздөрүнүн’ ‘буттарын’, ‘мыкты’, ‘илип’, ‘жатышат’]

lemmatize list of words and join *Lemmatized_output= ’ ‘.join ([lemmatizer.lemmatize(w) for w in word_list])*

print (lemmatized_output)

#> [Чаар жарганаттар өз бут мыкты илип жатышат.]

Мында берилген программа толук аткарылган жок, анткени ‘жарганаттар’- ‘жарганат’ деп, ‘илип’- ‘ил’ деп, биз күткөндөй өзгөргөн жок. Эгер лемматизациянын экинчи аргументи катары POS tag (сөз түркүмү боюнча талдоо) колдонсок, катаны оңдоого болот.

Айрым учурда бир эле сөздүн маанисине карай бир нече леммасы болот.

print(lemmatizer.lemmatize (“кат”, ‘v’)) (этиш)

#>кат

print(lemmatizer.lemmatize(“кат”, ‘n’)) (зат атооч)

#>кат

NLTK (табигый тил куралдары) да, сөз түркүмдөрү боюнча талдоо nltk.pos_tag шилтемеси аркылуу ишке ашат. Ал сөз тизмектерин эле эмес, бир гана сөздү да кабыл алат.

print(nltk.pos_tag(nltk.word_tokenize(sentence)

#> [(‘The’, ‘DT’), (‘striped’, ‘JJ’), (‘bats’, ‘NNS’), (‘are’, ‘VBP’), (‘hanging’, ‘VBG’), (‘on’, ‘IN’), (‘their’, ‘PRP$’), (‘feet’, ‘NNS’), (‘for’, ‘IN’), (‘best’, ‘JJS’)]

#> [The strip bat be hang on their foot for best]

#> [(‘Чаар’, ‘ADJ’), (‘ жарганаттар’, ‘NOUN_NOM’), (‘буттарын’, ‘NOUN_ACC’), (‘мыкты’, ‘ADJ’), (‘илип’, ‘VERB’), (‘жатышат’, ‘AUX’)]

#> [Чаар жарганат өз бут мыкты ил]

2. SpaCy лемматизатор

SpaCy Pythondогу табигый тилди өркүндөтүү үчүн акысыз, ачык булактар китепканасы болуп саналат. Ал – атайын иштелип чыккан тиркеме. Ошондой эле ал чоң көлөмдөгү текстти түшүнүп, иштеп чыгуучу башка тиркемелерди түзүүгө да жардам берет. Аны колдонуудан мурун, адегенде SpaCy тиркемесин орнотуп, ‘en’ (ен) моделин жүктөп алуу керек. [https://www.machinelearninghlus.com/nlp/lemmatization-examples-python/]

Сүйлөм= ”Чаар жарганаттар өздөрүнүн буттарын мыкты илип жатышат”

parse the sentence using the loaded ‘en’ model object ‘nlp’
doc=nlp(sentence)
extract the lemma for each token and join
“ “. join ([token.lemma_for token in doc])
#> ‘ Чаар жарганат –PRON(ат атооч)- бут жакшы ил’

Бул пакет Wordnet Lemmatizer сыяктуу эле, сөз түркүмдөрдү да туура энтектеп, лемматизациялап чыкты.

3. Pattern lemmatizer

Pattern Lemmatizer- бул көптөгөн пайдалуу ТТИ мүмкүнчүлүктөрүнө ээ болгон ар тараптуу модуль болуп саналат.

[https://www.machinelearninghlus.com/nlp/lemmatization-examples-python/]

сүйлөм=”Чаар жарганаттар өздөрүнүн буттарын илип, мыкты балыктарды жешти”
“ “.join([lemma(wd) for wd in sentence.split ()])
#>’Чаар жарганат өз бут ил мыкты балыктар же’

Ошондой эле мында ар бир сөз үчүн мүмкүн болгон лексеманы көрө аласыз.

#>lexeme for each word
[lexeme (wd) for wd in sentence.split()]
#> [[‘чаар’ ‘чаарала’ ‘ала ‘]
#> [‘жарганат’ ‘жарганаттар’ ‘жарганаттардын’ ‘жарганаттарга’ ‘жарганаттарды’ ‘жарганаттарда’ ‘жарганаттардан’ ‘жарганатка’ ‘жарганаттын’ ‘жарганатты’ ‘жарганатта’ ‘жарганаттан’ ‘жарганатым’ ‘жарганатың’ ‘’]
#> [‘өз’ ‘өзү’ ‘өзүм’ ‘өзүнүн’ ‘өзүнө’ ‘өзүн’ ‘өзүндө’ ‘өзүнөн’ ‘өздөрү’ ‘өздөрүнүн’ ‘өздөрүнө’ ‘өздөрүн’ ‘өздөрүндө’ ‘өздөрүнөн’]
#> [‘бут’ ‘буттар’ ‘буттардын’ ‘буттарга’ ‘буттарды’ ‘буттарда’ ‘буттардан’ ‘буттун’ ‘бутка’ ‘бутту’ ‘бутта’ ‘буттан’ ‘бутум’ ‘бутуң’ ‘бут’]
#> [‘ил’ ‘илип’ ‘илди’ ‘илген’ ‘илет’ ‘илишти’ ‘илишет’ ‘илишкен’]
#> [‘мыкты’ ‘жакшы’ ‘сонун’ ‘эң жакшы’]
#> [‘балык’ ‘балыктар’ ‘балыктардын’ ‘балыктарга’ ‘балыктарды’ ‘балыктарда’ ‘балыктардан’ балыктын’ ‘балыкка’ ‘балыкты’ ‘балыкта’ ‘балыктан’ ‘балыгым’ ‘балыгың’ ‘балыгы’]
#> [‘же’ ‘жеди’ ‘жейт’ ‘жейм’ ‘жейбиз’ ‘жейсиңер’ ‘жешти’ ‘жештилер’

4. Stanford CoreNLP [htpps://github.com)]

Stanford CoreNLP Java тилинде жазылган жана ал табигый тилди анализдөө каражаттарынын топтому менен камсыз кылат. Бул курал адам тилиндеги текстти киргизе алат. Ошондой эле сөздөрдүн негизги формаларын, компаниялардын, адамдардын ж.б.у.с. аталыштарын да тааный алат. Анын анализдери текстти жогорку деңгээлде жана түшүнүктүү колдонууну камсыз кылган негизги блокторду камтыйт.

[https://www.machinelearninghlus.com/nlp/lemmatization-examples-python/]

сүйлөм=“Чаар жарганаттар өздөрүнүн буттарын илип, мыкты балыктарды жешти”
corenlp_лемматизацияла сүйлөм формасын жана аны (лемма_тизмегине) бириктир #>
‘Чаар жарганат өз бут ил, мыкты балык

5. Түрк лемматизатору

Түрк лемматизатору түрк тилиндеги сөз өзгөртүүчү мүчөлөрдү анализдеген курал болуп саналат. Бул лемматизатор анализди аткарууда үч кадамды талап кылат. Алар: лексика, лемманы өзгөрткөн сөз өзгөртүүчү суффикстерди иштетүү жана суффикстердин жарактуу же жараксыз экенин текшерүү болуп эсептелет. Түрк тили үчүн мыкты лексика болуп түрк тили институтунун сөздүгү эсептелет. Бирок ал

маалымат катары онлайн колдонууга жеткиликсиз болгондуктан, альтернативдик жол менен пайдалануу керек. Альтернативдик жол менен пайдаланылуучу негизги сөздүк – Zargan Dictionary. Мында автор 1.3 млн сөз формаларын уңгулар менен камсыз кылган. Ушул файлдан алынган сөздөрдү лексика катары чогултууга болот.

Түрк тилинде лемманы өзгөртүүчү суффикстерди иштетүү (ablaut) деп аталат. Лемманын түзүлүшүн өзгөрткөн үч башка аблаут (уңгуну өзгөрткөн сөз өзгөртүүчү мүчөлөр) бар. Лемматизатор ушул өзгөрүүлөрдүн негизинде жаңы сөздөрдү жаратат. Бул өзгөрүүлөр: этиштин терс формасын, үнсүздөрдүн жумшаруусун, үндүүлөрдүн жакын (кууш) үндүүлөргө айлануусун, үндүүлөрдүн түшүп калуусун жана нөлдүк инфинитивди түзөт.

Этиштин терс формасын түзүү үчүн этишке (–ma –me) суффикстерин кошобуз.

[(https://github.com/akoksal/Turkish-Lemmatizer)]

M.: Gelmek->gelme**me**k

Үнсүздөрдү жумшартуу үчүн ар бир сөздү жумшартуучу вариант түзүү зарыл. Эгер леммалардын аягы [p, ç, t, k] жана башка үнсүздөр менен аяктаса, ага мүчө жалганганда [b, c, d, g/ğ] болуп өзгөрөт. Ал эми сөздүн аягы nk менен бүтсө, [g] болуп өзгөрөт. M.: kitap - kitabı, küçük - küçüğü, renk – rengi ж.б. [(https://github.com/akoksal/Turkish-Lemmatizer)]

Кууш үндүүлөргө айлануусу. Акыркы тыбышы ‘a’ же ‘e’ болгон сөздөргө “**yor**” уланганда, мүчөлөр ‘ı’, ‘i’ге өзгөрөт. Эрежеге баш ийбеген ‘demek’, ‘yemek’этиштеринен башкасынын баары өзгөрүүгө учурайт.

Үнсүздөрдүн түшүп калуусу түрк тилиндеги эки муундан турган айрым сөздөргө үндүү менен башталган мүчөлөр уланганда, сөздүн акыркы муунундагы үндүү түшүп калат. M.: zehr - zehri, oğul – oğlu. [(https://github.com/akoksal/Turkish-Lemmatizer)]

Жогоруда белгиленген процесстердин Python пакетинде аткарылышы төмөндөгүдөй жүргүзүлөт. Адегенде сөздүктү zargan менен топтоо үчүн train lexicon.py кыймылын иштетип, 2-бөлүмдөгү өзгөрүүлөрдү адаптация кылуу зарыл [Pythons trainLexicon.py]. Бул буйруктан кийин демейки код 2 бөлүмүндөгү кадамдарды колдонуп, өзгөртүлгөн сөздүктү түзүү үчүн, маалымат топтомундагы zargan.pkl файлы колдонулат. Бул кадамдан кийин revisedDict.pkl файлы түзүлөт. Мындан соң, 3-бөлүмдө lemmer.py файлын жана текшергичти иштетип, сөздөрдүн каалаган леммасын текшерүүгө болот. [(https://github.com/akoksal/Turkish-Lemmatizer)]

Python3 lemmatizer.py ağacı

İnput is “ağacı“ output”ağaç“

ağaç_1

ağa_1

ağ_1

a_1

Мында ağacı сөзүнүн туура леммасы ağaç. Төмөндөгү бир нече мисалдарды карайлы:

[(https://github.com/akoksal/Turkish-Lemmatizer)]

Көз айнекчилер, көз айнекчи, көз айнеги > көз айнек.

>>>lemmatization=lemmatizer.lemmatize_text

(Анын эмгегинин акыбети кайтты.)

>>>for(sentence, lemmas) in lemmatization

>>>print.sentence

Анын: [‘ага’, ‘ал’]
Эмгегинин: [‘эмгеги’, ‘эмгек’]
Акыбети: [‘акыбет’]
Кайтты: [‘кайт’]

6. Кыргыз тилиндеги сөздөрдү лемматизациялоо жана андагы кыйынчылыктар

Төбөсү кызыл, чокчойгон быжыгыр көрпө бөркү даана ажыратылат, саймалуу чоң чачыгы селкилдейт. [(https://corpora.clarin-d.uni-saarland.de/cqpweb/kyrgyz_20190418) "Сынган кылыч" романынан].

Файзабад да аңгырап бош калган. Ачыгы ачык, чачыгы чачык. [(https://corpora.clarin-d.uni-saarland.de/cqpweb/kyrgyz_20190418) "Сынган кылыч" романынан]

print (lemmatizer.lemmatize(‘чачыгы’, “n”)
#>чачык
print (lemmatizer.lemmatize(‘чачык’, “v”)
#>чачык

Кайра жаачу булуттай, Каары бетине айланып...

[(https://corpora.clarin-d.uni-saarland.de/cqpweb/kyrgyz_20190418) "Манас" эпосунан]

Тарткан жаасын карасаң, Болчу кылдай чоюлду... [(https://corpora.clarin-d.uni-saarland.de/cqpweb/kyrgyz_20190418) "Манас" эпосунан]

print (lemmatizer.lemmatize (‘жаачу ‘, “v”))
#>жаа
print(lemmatizer.lemmatize (‘жаасын ‘, “n”))
#>жаа

[(https://corpora.clarin-d.uni-saarland.de/cqpweb/kyrgyz_20190418) "Сынган кылыч" романынан]

“Исхак өз туусуна бириккен кошуундун санын тактады.”

[(https://corpora.clarin-d.uni-saarland.de/cqpweb/kyrgyz_20190418) "Сынган кылыч" романынан]

“Кап ортого тактадан бир секи орнотулган.”

print (lemmatizer.lemmatize (‘тактады’, “v”)
#>такта
print (lemmatizer.lemmatize(‘тактадан’, “n”)
#>такта

Демек, кыргыз тилиндеги бир эле сөз бир нече маанини билдиргендиктен, лемматизациялоодо төмөнкүдөй көйгөйлөрдү жаратары мисалдардан көрүнүп турат. Эгер кыргыз тилине лемматизация процесси жүргүзүлсө, мындан тышкары да бир нече тоскоолдуктар болот. Алсак,

1) айрым сөздөргө мүчөлөр уланганда, сөздүн аяккы тыбышынын өзгөрүшү. Мисалы: китеп+ым=китебим, табак+ы=табагы, бак+ыл= багыл

2) аягы –ск, -нк, -фть, -кт, -нг, -нд, -мн менен бүткөн сөздөргө сөз өзгөртүүчү мүчө жалганганда, уңгу менен мүчөнүн ортосуна үндүүлөрдүн кошулушу.

М.: Минск+нын=Минскинин, акт+лар=актылар ж.б.у.с.

Булар англис жана түрк тилиндеги лемматизацияны окуп, изилдеп кыргыз тилине

салыштыргандан кийинки табылган көйгөйлөр болду. Эгер мындан ары да изилдөөнү илгерилетсек, дагы бир нече көйгөйлөр чыгары анык. Мындан улам, бул сыяктуу маселелерди чечүүнүн жолдорун изилдеп табуу керек деген ойго келебиз.

Корутунду

Жалпысынан, киришүүдө лемматизация табигый тил куралынын жана айрым Python пакеттеринин жардамы менен сөздөрдүн мүчөлөрүн жойгон алгоритмдик процесс экенин билдик. Ал эми макаланын экинчи, үчүнчү бөлүмдөрүнөн лемматизация жана анын стеммингден айырмасы жөнүндө толук маалымат берилди. Төртүнчү бөлүмдө бул анализдин Python пакеттери менен бирге колдонулушу тууралуу берилсе, бешинчиде түрк лемматизатору жөнүндө айтылды. Ошондой эле алтынчы бөлүмдө кыргыз тилиндеги сөздөрдү лемматизациялоо жана андагы көйгөйлөр тууралуу баса белгиленди. Жыйынтыктап айтканда, бул макалада англис жана түрк табигый тил иштетүү процессиндеги лемматизацияны изилдеп, аны кыргыз тилине салыштырдык. Лемматизация кыргыз тили үчүн жаңы термин болгону менен, ал морфологиялык талдоонун, башкача айтканда, уңгу жана мүчөгө ажыратуунун кол менен эмес, компьютер тарабынан аткарылган процесси болуп саналат. Муну менен бирге, түрк жана кыргыз тилиндеги лемматизация көйгөйлөрү бири-бирине окшош болору аныкталды жана макаладагы мисалдардын айрымдары жаңы түзүлгөн кыргыз тилинин корпусунан алынды. Бул эмгек кыргыз тили үчүн лемматизациянын башталышы гана болгондуктан, аны алдыга илгерилетүүдө дагы көптөгөн көйгөйлөр, тоскоолдуктар болот жана аларды чечүүнүн жолдору да изилденет.

Шилтемелер:

1. https://nlp.stanford.edu
2. https://blog.bitext
3. https://www.geeksforgeeks.org
4. https://towardsdata.com
5. https://www.machinelearningplus.com
6. https://corpora.clarin-d.uni-saarland.de/cqpweb/kyrgyz_20190418
7. https://github.com/akoksal/Turkish-Lemmatizer
8. Акынбекова А.У. Кыргыз тили Avrasya press, www avrasyapress.com
9. http://vestnik.krsu.edu.kg/archive/154/6520