УДК 004.421.6

Эгембердиева Жылдыз Сраждиновна,
аспирант,
Ошский технологический университет им. М.М. Адышева
Эгембердиева Жылдыз Сраждиновна,
аспирант,
М.М. Адышев ат. Ош технологиялык университети
Egemberdieva Zhyldyz Srazhdinovna,
graduate student,
Osh Technological University named after M.M. Adysheva

Көчкөнбаева Бүажар Осмоналиевна,
к.т.н., доцент,
Ошский государственный университет
Көчкөнбаева Бүажар Осмоналиевна,
т.и.к., доцент,
Ош мамлекеттик университети
Kochkönbayeva Bүazhar Osmonalievna,
Candidate of Technical Sciences, Associate Professor,
Osh State University

Сатыбаев Абдуганы Джунусович,
профессор,
Ошский технологический университет им. М.М.Адышева
Сатыбаев Абдуганы Джунусович,
профессор,
М.М.Адышев ат. Ош технологиялык университети
Satybaev Abdugany Dzhunusovich,
professor,
Osh Technological University named after M.M. Adysheva

ТАБИГЫЙ ТИЛДЕГИ ТЕКСТТЕРДИН ҮСТҮНӨН ИШТӨӨДӨ СӨЗ ФОРМАЛАРЫНА АНАЛИЗ ЖАНА СИНТЕЗ ЖҮРГҮЗҮҮНҮН АЛГОРИТМИН ТҮЗҮҮ

Аннотация. Бул макалада табигый тилдеги тексттерди сөз формаларына ажыратуунун алгоритми жана пайда болгон сөздүктөгү сөз формаларына уланган аффикстерге анализ жана синтез жүргүзүүнүн алгоритми каралды. Ар бир түзүлгөн алгоритм табигый тилдин эрежелерин эске алуу менен иштелип чыкты.

Негизги сөздөр: сөз формалары, формалдуу тилдер, көптүктөр, аффикс,маалыматтар базасы

РАЗРАБОТКА АЛГОРИТМА АНАЛИЗА И СИНТЕЗА СЛОВФОРМ ПРИ РАБОТЕ НАД ТЕКСТАМИ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

Аннотация. В данной статье рассматривается алгоритм разделения текстов естественного языка на словоформы и алгоритм анализа и синтеза аффиксов словоформы в полученном словаре. Каждый созданный алгоритм разрабатывался с учетом правил естественного языка.

Ключевые слова: словоформы, формальные языки, множество, аффиксы, базы данных

CREATING AN ALGORITHM FOR ANALYSIS AND SYNTHESIS OF WORD FORMS ON TEXTS IN NATURAL LANGUAGE

Abstract. This article discusses an algorithm for dividing natural language texts into word forms and an algorithm for analyzing and synthesizing word form affixes in the resulting dictionary. Each created algorithm was developed taking into account the rules of natural language.

Key words: word forms, formal languages, set, affixes, databases

Киришүү. Маселенин коюлушу

Табигый тилдерде компьютерге кандайдыр бир маселени чечүү тапшырмасын бере албайбыз. Ошондуктан табигый интеллект менен компьютердин ортосунда ар түрдүү формалдык тилдер иштелип чыккан. Мисалы, кагаз бетине түшүрүлгөн объекттин иммитациалык моделин текшерүү үчүн Matlab же Python формалдык тили иштелип чыккан. Ошентип Хомскийдин окууларына таянсак формалдык тилдер – табигый тилдердин математикалык модели болуп эсептелет. Ал эми ар бир формалдык тил өзүнүн формалдуу грамматикасына ээ болот.

Жогоруда айтылгандардын негизинде кыргыз тилинин мисалында табигый тилдин формалдык грамматикасын түзүп көрүү чечимине келдик. Анткени табигый тилдин формалдуу түрүн түзүү менен, ошол тилде түшүнө турган жасалма интеллекттин биринчи кадамдарын жасаган болобуз.

Тилдин алфавити деп чектүү, элементтери тамгалар же белгилер болгон курук эмес көптүктү айтабыз. Эгерде T алфавиттеги тамгаларды удаалаш жазсак, анда T алфавитиндеги тамгалар чынжырчасы келип чыгат б.а. $T=\{А\text{-}Я,а\text{-}я\}$. Алынган чынжырчанын узундугу α анда камтылган символдордун санына барабар жана аны |α| түрүндө белгилеп алалы. Курук көптүк бир да символ камтыбайт жана e менен белгиленет.

Эгерде α жана β – эки символдор чынжыры болсо, анда $\omega=\alpha\beta$, α жана β чынжырларынын конкатенциясы деп аталат. Ар кандай a символу жана k $(k\geq 0)$ үчүн анын конкатенациясы a^k деп белгиленет б.а.

$a^0=e$, $a^1=a$, $a^{k+1}=a^k a$ $(k>=1)$ болот.

Конкатенация операциясынын жардамында ар түрдүү чынжырчалар түзүлөт (сөздөр, жолчолор). Чынжырчалардын үстүнөн кайрадан конкатенация операциясын жүргүзүүгө болот жана чынжырча алынат.

Эгерде T- алфавит болсо, анда бул алфавиттеги чынжырлардын (сөздөрдүн) көптүгү төмөнкүдөй аныкталат:

$$T^* \leftrightarrows \bigcup_{k=0}^{\infty} T^k$$

мында $T^0 \leftrightarrows \{\varepsilon\} - 0$ узундуктагы чынжырчалардын көптүгү.

$T^k \rightleftharpoons \underbrace{T \cdot T \cdot \ldots \cdot T}_{k}$ - k $(k>=1)$ узундуктагы чынжырчалар көптүгү. T көптүгү бардык сөздөрдүн көптүгүн түзөт. Бул макалада, биз, кыргыз тилинин мисалында ушул T көптүгүндөгү сөздөргө анализ жана синтез жасоону мак-

сат кылып алдык жана бул маселенин коюлушу болду.

Табигый тилдеги тексттен сөздөрдү бөлүп алуунун алгоритми

Кыргыз тилинде сөз түрлөрү уңгунун $T=\{t_1, t_2,...,t_n\}$ жана аффикстердин $M=\{m_1, m_2,...,m_n\}$ (мүчөлөр) конкатенациялоо жолу аркылуу жасалат. Мында ар бир аффикске көптөгөн семантикалык сыпаттар менен байланышкан жана аффикстерди кошуу ирети тартиби менен аныкталат. Мисалы, зат атоочтор үчүн сөздүн негизине башында көптүк түрдүн мүчөсү кошулат, андан кийин илик мүчө, андан ары жөндөмө мүчөсү келет жана андан кийин гана жактаманын түрүнүн мүчөсү (зат атоочтун жандууларына гана кошулат)[1].

Жаңы сөз түрлөрү башкы формалардын морфологиялык жана семантикалык сыпаттарын эске алуу менен жасалат: сөздүн башкы формасы; андан кийин, солдон оңго жылып, тиги же бул мүчөнү кошуу үчүн сөздүн башкы формасынын акыркы тамгасынын (акыркы үн) категориясы (үндүү жана үнсүздөр ж.б.у.с.) аныкталат.

Курамды аныктоонун жалпы морфологиялык формасы мындайча көрүнөт: Уңгу (корень) + мүчө (окончание) [2].

Кыргыз тилиндеги сөздөргө мүчөлөрдүн улануу модели төмөнкү мисалда көрсөтүлгөндөй ишке ашат:

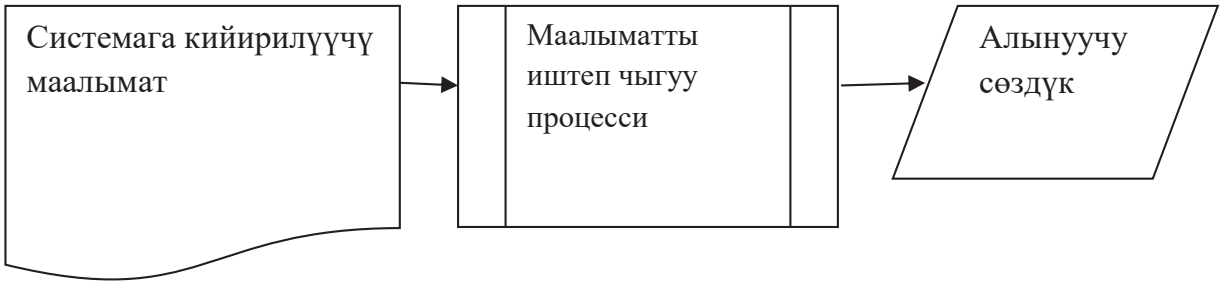
1. If w_n*='а'||'ы' then,* $M=\{m_1,m_2,m_3\}$*;* m_2*='ы';*

2. If w_n*='о'||'у' then* $M=\{m_1,m_2,m_3\}$*;* m_2*='у';*

3. If w_n*='э'||'u' then,* $M=\{m_1,m_2,m_3\}$*;*m_2*='u';.*

4. If w_n*='ө'||'ү' then,* $M=\{m_1,m_2,m_3\}$*;* m_2*='ү'.*

Белгилүү болгондой, морфемалар тилдин эң кичине маани берүүчү (семантикалык) бирдиги болуп саналат, алардан сөздүн формасы, андан ары, ошого жараша, лексема да түзүлөт. Мындай сөз формаларын туура түзүп берүүчү жасалма интеллектти түзүүгө чейин табигый тилде жазылган текстти анализдөөнү туура көрдүк б.а. төмөнкү системанын алгоритмин иштеп чыктык (1-сүрөт).

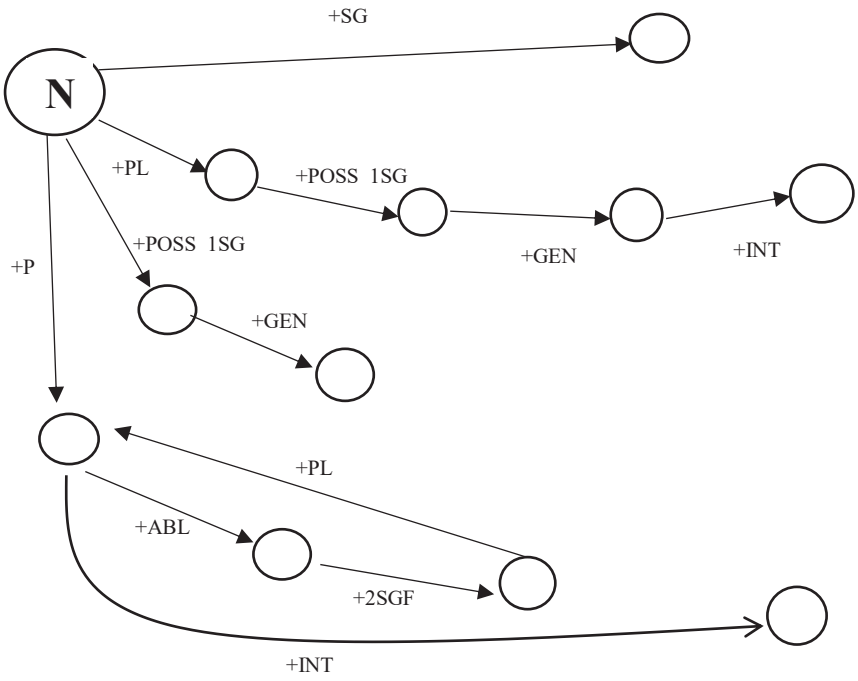


1-сүрөт. Системанын иштөө алгоритми

Бул алгоритмдин негизинде Python программалоо тилинде программасы иштелип чыкты.

Сөздүктөгү сөздөргө анализ жана синтез жүргүзүү алгоритми

Сөздүктөгү сөздөрдү кароодо мисал катары атооч сөздөрдү бөлүп алдык. Атооч сөз формалары сөзгө мүчөлөрдүн төмөнкү фрейм-модели боюнча уланышы аркылуу ишке ашырылат (2-сүрөт). Көрүнүп тургандай, моделдин түйүндөрү атооч сөздөрдүн морфологиясына тиешелүү ар кандай абалдары көрсөтсө, түйүндөрү байланыштырып турган өтмөктөр (жебечелер) атооч сөздөрдүн морфологиясына тиешелүү конкреттүү категорияларды көрсөтөт. Ал эми N – атооч сөздүн (noun) сөздүктөгү формасы (негизи).



2-сүрөт. Атооч сөз формаларын уюштуруунун фрейм-модели

Жогорудагы моделдин негизинде сөздүктөгү сөздөргө анализ жасоо үчүн *word (x)* предикатын ала турган болсок, ал х объекти менен мүчөлөр көптүгүнүн ортосундагы байланышты камсыз кылат, башкача айтканда, төмөнкү предикаттар көптүгүн алабыз [3]: [word (N,SG), word (N,P, POSS_1SG,GEN,INT), word (N, POSS_1SG,1GEN), word (N, PL, ABL, 2SGF, PL, INT)]. Бул жерде объекттердин ортосунда түз же түз эмес байланыштар орун алуусу мүмкүн.

Мисал катары *китеп* сөзүн алсак, жогорудагы модель боюнча төмөнкү тизме алынат:

китеп+SG

китеп+PL+ POSS_1SG + GEN + INT

китеп+ POSS_1SG+1GEN

китеп+PL+ABL+2SGF+PL+INT

Жогорудагы мисалдардын негизинде биз төмөнкүдөй алгоритмди сунуштайбыз:

1) Негиздин бардык варианттары табылат жана мүчөлөр алынып салынат.

2) Ар бир уңгу варианты үчүн эң узунунан баштап, уңгулар тизмесинен бинардык издөө жүргүзүлөт. Эгерде бул тизмеде уңгу варианты жок болсо, анда «эң жакын» сөздүктөгү уңгу ушундай жол менен табылат. Биринчи «жакын» уңгунун орду жана анын окшоштук өлчөмү - уңгудагы дал келген символдордун саны жана аяктоо узундугу эске алынат.

3) Уңгунун бардык варианттары үчүн төмөнкүлөр аткарылат: Окшоштуктун өлчөмү бирдей болгон бардык лексемалар үчүн морфологиялык анализ аткарылат. Эгерде уңгу варианты “жакын” сөздүк уңгуларынын бирине да дал келбесе, анда бул уңгу варианты бар талданган сөз сөздүктө жок дегенди билдирет.

4) Андан ары бөлүнүп алынган сөз формасы үчүн тексттеги аффикстердин уланууу тизмеси текшерилет. Эгерде туура болсо 5-кадамга өтөбүз, антпесе катанын себептери көрсөтүлөт.

5) Анализдин аягы

Ошентип, табигый тилдеги текст үчүн морфологиялык анализатордун иштөө принциптерин кароодо төмөнкүдөй жагдайлар келип чыгат:

1. Киргизилген текстти сөз формаларына бөлүп алуу.

2. Андан соң сөз формаларын лемматизациялоо, атап айтканда, сөз формасын

сөздүн сөздүктөгү формасына айландыруу.

3. Сөз формасын уюштурган уланды мүчөнү же мүчөлөрдүн тизмегин бөлүп алуу.

4. Уланды мүчөлөр тизмегин андан ары жиктөө жана ар бир мүчөнүн тиешелүү морфологиялык белгилерин аныктоо.

Мисалы: балдар сөзүн талдоодо морфологиялык анализатор сөздүктөгү балдар формасы аркылуу ал бала сөзүнөн уюшулганын жана –лар мүчөсү жалганганда сөздүн соңку «а» тыбышы түшүп калганын аныктайт. Ал эми китебим формасында китеп сөздүктөгү сөз деп, анын соңку «п» тыбышы жумшарып «б» тамгасына өтүп кеткен деп аныктоо зарыл.

Маселени чечүүнүн эки жолу бар: биринчиси кийирилген сөздү оңдон солго карай анализдөө, экинчиси солдон оңго карай анализдөө ыкмасы. Айтылган ыкмалардын алгоритмдерин карап көрсөк, төмөнкүдөй жыйынтыкка ээ болобуз.

«Оңдон солго» ыкмасында сөздөр оң жактан баштап мүчөлөргө ажыратыла баштайт. Эгерде илимий текстте сөздөр орточо 7-10 сандагы тамгадан турат десек, анда морфологиялык анализатор жок дегенде 5-6 жолу сөздүккө (компьютердин эсине) кайрылуу жасайт.

«Солдон оңго» ыкмасын карап көрсөк, бул учурда кийирилген сөздүн сөздүктө жолуккан эң жакын негизин табуу амалы жүрөт да, калган бөлүгү мүчөлөр болуп, алар өзүнчө сөздүктөн салыштыруу аркылуу аныкталат. Бул учурда да цикл бир канча жолу кайталанып, ошол эле учурда кайталоо ийгиликсиз болуп калуу ыктымалдуулугу да жок эмес.

Жогоруда айтылгандардын негизинде морфологиялык анализде компьютердин эсине кайрылуу көп жолу кайталанары талашсыз. Ошондуктан компьютердин эсине кайрылууну азайтуу маселеси да актуалдуу болуп турат. Албетте, биз бул учурда сөздүктү түзүү маселесин, башкача айтканда, маалыматтар базасын башкарууну карообуз туура болот.

Жыйынтык

Жогоруда каралгандардын негизинде сөз формаларына аффикстерди улоонун эрежесин карап чыктык. Алардын тууралыгына анализ жана синтез жасоо менен табигый тилдин негизинде туура формалдык тилди жана ошондой эле жасалма интеллектти жасоого мүмкүн экендигин көрсөтө алдык. Ошондой эле анализ жана аффикстерге синтез жасоонун алгоритмин карап чыктык.

Адабияттар

1. *И. А. Батманов,* Части речи в киргизском языке: материалы к стандартной схеме морфологии киргизского языка. Фрунзе: Киргизгосиздат, 1936. – 48 с.

2. *И. Абдувалиев, Т. С. Садыков,* Современный кыргызский язык (Морфология) – Бишкек: Айбек, 1997. – 296 с.

3. *Б.О. Кочконбаева, Т. С. Садыков,* Модель морфологического анализа кыргызского языка - Издательство Академии наук Республики Татарстан – Казань, 2017