Улсын бүртгэлийн
дугаар: . . . . . .

Нууцын зэрэглэл: А

Төсөл хэрэгжүүлэх
гэрээний дугаар:
ШуСс-2017/04

Аравтын бүрэн
ангиллын код

ШИНЖЛЭХ УХААНЫ АКАДЕМИ
МАТЕМАТИК, ТООН ТЕХНОЛОГИЙН ХҮРЭЭЛЭН

МАШИН СУРГАЛТЫН АРГЫГ КИРИЛЛ, МОНГОЛ БИЧГИЙН АЛДАА ЗАСАХ, БИЧВЭР ХООРОНД ХӨРВҮҮЛЭХЭД АШИГЛАХ НЬ

СУУРЬ СУДАЛГААНЫ ТӨСЛИЙН ТАЙЛАН
(2017-2019)

Төслийн удирдагч: Дуламрагчаагийн Уганбаатар, Доктор (Ph.D), Дэд профессор

Санхүүжүүлэгч байгууллага: Шинжлэх ухаан, технологийн сан

Захиалагч байгууллага: Боловсрол, соёл, шинжлэх ухаан, спортын яам

Тайлан өмчлөгч: Математик, тоон технологийн хүрээлэн
Улаанбаатар, УБ-54, 210351,
Энхтайваны өргөн чөлөө – 54Б,
И-мэйл: imdt@mac.ac.mn
Утас: (976)-11-458090

Улаанбаатар хот
2019

Улсын бүртгэлийн
дугаар: . . . . . .

Нууцын зэрэглэл: А

Төсөл хэрэгжүүлэх
гэрээний дугаар:
ШуСс-2017/04

Аравтын бүрэн
ангиллын код

ШИНЖЛЭХ УХААНЫ АКАДЕМИ
МАТЕМАТИК, ТООН ТЕХНОЛОГИЙН ХҮРЭЭЛЭН

МАШИН СУРГАЛТЫН АРГЫГ КИРИЛЛ, МОНГОЛ БИЧГИЙН АЛДАА ЗАСАХ, БИЧВЭР ХООРОНД ХӨРВҮҮЛЭХЭД АШИГЛАХ НЬ

СУУРЬ СУДАЛГААНЫ ТӨСЛИЙН ТАЙЛАН
(2017-2019)

Төслийн удирдагч:	Дуламрагчаагийн Уганбаатар, Доктор (Ph.D), Дэд профессор
Санхүүжүүлэгч байгууллага:	Шинжлэх ухаан, технологийн сан
Захиалагч байгууллага:	Боловсрол, соёл, шинжлэх ухаан, спортын яам
Тайлан өмчлөгч:	Математик, тоон технологийн хүрээлэн Улаанбаатар, УБ-54, 210351, Энхтайваны өргөн чөлөө – 54Б, И-мэйл: imdt@mac.ac.mn Утас: (976)-11-458090

Улаанбаатар хот
2019

Удирдагч:	Д.Ууганбаатар	МТТХ-ийн ЭША, доктор
Төсөлд оролцогчид:	М.Хүрэлхүү	МТТХ-ийн ЭША, магистр
	И.Бямбасүрэн	МТТХ-ийн ЭША, докторант
	М.Мархамет	Оюу Толгой ХХК, магистр
	Ч.Содоо	МТТХ-ийн ЭША, магистр
	Э.Батзаяа	МТТХ-ийн ЭША
	Т.Пүрэвсүрэн	ХЗХ-ийн ЭША, доктор
	Б.Нэргүй	МТТХ-ийн ЭША, доктор
	М.Тунгаацэцэг	МТТХ-ийн туслах ажилтан

ГАРЧИГ

РЕФЕРАТ

Тайлангийн нэр: “Машин сургалтын аргыг кирилл, монгол бичгийн алдаа засах, бичвэр хооронд хөрвүүлэхэд ашиглах нь”

Тайлан 99 хуудас, 65 зураг, 25 хүснэгттэй. Улаанбаатар хот, 2019 он

Тайланг өмчлөгч: ШУА-ийн Математик, тоон технологийн хүрээлэн

УДИРТГАЛ

Зорилго: Өнөөдөр мэдээллийг автоматаар боловсруулах, хэрэглэхийн ач холбогдол улам бүр их болж, энэ чиглэлийг хөгжүүлэхгүйгээр урагш алхах боломжгүй болжээ. Дэлхий нийтээрээ мэдээллийн эрин зуунд шилжиж олон улс орон өөрийн хэл, бичгийг компьютерээр боловсруулах, зүй тогтлыг нь компьютерт таниулахаар их хүч хөдөлмөр, их хөрөнгө зарцуулж, түүнийхээ хэрээр дэлхийн хөгжилтэй хөл нийлэх болсон. Манай улсын хувьд энэ талын судалгаа, шинжилгээний ажил эхлэлийн байдалтай байна.

Иймээс тооцоолох хэл шинжлэлд өргөн хэрэглэгдэж буй үг зүйн загварыг судалж, монгол хэлний хэл шинжлэлд хэрхэн хэрэглэж болохыг харуулах энэхүү сул орон зайг бага ч болсон нөхөж монгол хэлийг цахим орчинд боловсруулахад зориулсан онол арга зүй, практик арга аргачлалыг боловсруулах.

Цахим орчинд монгол хэл, бичгийн хэрэглээг нэмэгдүүлэх зорилгоор монгол хэл, кирилл болон монгол бичгийг компьютерээр боловсруулах чиглэлд олон судалгаа шинжилгээ хийх, холбогдох программ хангамж хөгжүүлэхэд шаардагдах онолын судалгаа хийж гүйцэтгэх, зарим нэгэн программ хангамж боловсруулж турших, хэрэглээг нэмэгдүүлэх, тэдгээрийг цахим орчинд нэвтрүүлэхэд хувь нэмэр оруулах.

Мэдээллийн технологийн хөгжлийн шинэ чиг хандлага болж буй “машин сургалт”-ын төрөл бүрийн технологи, арга зүйг монгол хэлний цахим боловсруулалтанд ашиглаж компьютер хэл шинжлэлийн судалгааныхаа үр дүнг сайжруулан холбогдох туршилтыг хийж гүйцэтгэх.

Судлагдсан байдал: Одоогийн байдлаар манай их дээд сургууль болон судалгаа шинжилгээний байгууллагуудад зарим түвшинд тодорхой ажлууд хийгдэж байгаа боловч бусад орны хүрсэн түвшинтэй харьцуулахад чамлалттай, нэгдсэн тогтолцоонд орж чадаагүй байна.

Хэдийгээр компьютер хэл шинжлэлийн чиглэлээр тодорхой судалгаанууд хийгдэж зарим нэгэн үр дүнгүүд гарч байгаа боловч хэдэн толь бичгээс өөр нийт хэрэглэгчид шууд ашиглах хэрэглээний программууд алга байна. Монгол хэлний материалыг ашигладаг төрөл бүрийн программ хангамжууд зохиогдсоор байгаа нь нэг үеэ бодвол сайшаалтай боловч тэдгээр программ хангамжид хэл боловсруулалтын суурь судалгаа, технологийн шийдэл дутагдаж байгаа билээ.

Өмнө нь бид энэ чиглэлээр монгол хэлний үг зүйн болон хөрвүүлэх систем, монгол хэлний дүрмийг хоёр түвшинт морфологид тулгуурлан загварчлах болон

монгол бичгийн хэвлэмэл материалыг таних талаар тодорхой хэмжээний ажлуудыг хийж гүйцэтгэж байсан болно.

Нэг хэлний хоёр бичгийн бичвэрийг хооронд нь хөрвүүлэхэд заавал уламжлалт машин орчуулгын аргыг хэрэглэх шаардлагагүй. Бид хоёр бичгийн хооронд үг үгээр нь харгалзах утгыг орлуулах замаар шууд хөрвүүлж болно гэсэн үг юм. Дэлхийд хоёр хэлний хооронд орчуулахад толь үүсгэх арга, дүрмэнд суурилсан арга болон статистик өгөгдөл дээр суурилсан аргуудыг голчлон хэрэглэдэг.

Кирилл болон монгол бичгийн хоорондох хөрвүүлгийн хувьд үүсмэл хэлэнд голчлон хэрэглэгддэг дүрмэнд суурилсан аргыг хэрэглэхэд илүү оновчтой болох нь бидний судалгааны ажлаас харагдлаа. Энэ арга нь хэлний үг зүйн дүрмийг загварчлах болон хэлний өгүүлбэр зүйн дүрмийг загварчлах гэсэн хэсгүүдээс бүрддэг. Кирилл болон монгол бичиг нь өгүүлбэр зүйн бүтцийн хувьд адилхан тул монгол хэлний эдгээр хоёр бичгийн хувьд үг зүйг нь загварчлахад хангалттай юм.

Шинжлэх ухааны ач холбогдол: Монгол хэлний кирилл болон монгол бичгийн асуудлуудыг цахим орчинд шийдвэрлэх зарим нэгэн арга зүй бий болж холбогдох хэрэглээний программ хангамжууд боловсруулагдаж нэвтрэх боломжтой болохоос гадна шинээр кирилл болон монгол бичгийн бичвэрийг боловсруулахад чиглэгдсэн олон программ хангамж бүтээн боловсруулах, хэрэглээнд нэвтрүүлэх ажлын онолын үндэслэл бий болно.

Гарсан үр дүн дээр үндэслэн цаашид хэлзүй шалгуур (grammar checking), бичвэр хураангуйлах (text summarisation), асуултад хариулах (question answering), тооцооллын утгазүй (computational semantics), машин орчуулга (machine translation), ярианаас бичвэрт, бичвэрээс ярианд хөрвүүлэг хийх (conversion of speech to text and text to speech), ярианы хэлний харилцан ярих систем (spoken language diolog systems) зэрэг системүүдийг хөгжүүлэх боломж бүрдэнэ.

Сэдэвт ажлаар дэвшүүлэх таамаглал, бүтээл туурвилын чиглэл: Сэдэвт ажлыг олон жилийн баялаг түүхтэй монгол хэл судлалыг орчин үеийн технологитой уялдуулж, монголын компьютер хэл шинжлэлийн арга техникийг (арга зүйг) боловсруулахад хувь нэмэр оруулах чиглэлээр гүйцэтгэнэ. Тухайлбал Монгол хэлний кирилл ба уламжлалт монгол бичгийн бичвэрээр өгөдлийн санг бүрдүүлж, холбогдох программ хангамжуудыг туршин боловсруулах, монгол хэлний судалгаанд цахим техник технологийг нэвтрүүлэх, монгол хэл судлалын уламжлалт аргыг өргөжүүлэн хөгжүүлж монголд компьютер хэл шинжлэлийн судалгааг хөгжүүлэн шинэ түвшинд гаргах, монгол хэлний өв соёлыг хамгаалахад тодорхой хувь нэмэр оруулах болно. Цаашид машин сургалтын аргуудыг монгол хэлний цахим боловсруулалтанд ашиглах судалгааны эх суурь тавигдсанаар кирилл болон монгол бичгийн цахим боловсруулалтанд шинэ дэвшил гарч бидний шинээр боловсруулах программ хангамжийн үр дүнгүүд сайжирна. Бид судалгааныхаа ажилд зориулан тусгай зориулалтын кирилл болон монгол бичгийн тус бүрийнх нь онцлогийг хадгалж чадсан өгөдлийн сангуудыг зохион байгуулж бүрдүүлэх ба энэ нь компьютер хэл шинжлэлийн талаар цаашид бидний хийх олон ажлын суурь болно.

Төслийн хүрээнд хийгдсэн ажлууд: Судалгааныхаа ажилд зориулан кирилл болон монгол бичгийн тус бүрийнх нь онцлогийг хадгалж чадсан тусгай зориулалтын өгөдлийн сангуудыг зохион байгуулсан. Энэхүү өгөгдлийн санд үгийн монгол, кирилл бичгээрх зохистой бичлэгийг оруулсан бөгөөд хоёр бичгийн зөв бичих дүрмийн тогтолцоог компьютерт таниулах аргачлалыг боловсруулан багтаасан зэрэг нь шинэлэг тал болно. Компьютер хэл шинжлэл, монгол хэлний кирилл болон монгол бичгийн бичвэрийг цахимаар боловсруулах чиглэлээр цаашид хийж гүйцэтгэх бидний хийх олон ажлын суурь нь төслийн ажлаар бүрдүүлсэн энэ сан байх болно. Байгуулсан өгөгдлийн сангийн бүтэц, тоо хэмжээний тухай тайлангийн 1-р бүлгээс, өгөгдлийн сан бүрдүүлэх, бүрдүүлсэн өгөгдлийн сангаа удирдах системийн тухай 3-р бүлгээс тус тус дэлгэрүүлэн уншина уу.

Бидний байгуулсан өгөгдлийн сан нь цаашид монгол хэлийг компьютерээр боловсруулах (үг зүйн хувилал, өгүүлбэр зүйн задлагч, кирилл ба уламжлалт монгол бичгийн бичвэр хооронд хөрвүүлэгч болон зөв бичих) ажлуудыг хийж гүйцэтгэх үндэс суурь нь болно.

Төслийн дараагийн шатны ажилдаа компьютерын тусламжтайгаар кирилл ба монгол бичгээр бичсэн үгэнд үг зүйн шинжилгээ хийх, үгийг зөв бичгийн дүрмийн дагуу нөхцөлөөр хувилгах аргыг тодорхойлохыг оролдлоо. Энэхүү нийлмэл процесс нь компьютер хэл шинжлэлийн ухааны компьютерын үг зүйн салбарт хамаарах ба эх хэлийг компьютерээр боловсруулах эхний ажлуудын нэг билээ. Компьютерын үг зүйн шинжилгээ нь үгийн хэлбэр үүсгэх /generation/, үгийг бүтцээр задлах /analysis/ гэсэн үндсэн хоёр үйлдэлтэй.

Компьютерын үг зүй (computational morphology)-н түвшинд үгийн хувиллыг таниулах буюу үгийг хувиллаар нь задлах болон үүсгэх хоёр чиглэлт үйлдлийг төгсгөлөг төлөвт автомат (finite state automata)-д үндэслэх нь үр дүнтэй гэж үзсэн. Энэ зорилгийнхоо хүрээнд төгсгөлөг автомат, хоёр түвшинт үг зүйн онолын үндэс болон монгол хэлний үг зүйг судалж түүнийхээ үндсэн дээр монгол хэлний үг зүйд боловсруулалт хийсэн.

Бид байгуулсан сандаа түшиглэн монгол хэлний үг зүйн түвшинд үгийг хувиллаар нь задлах болон үүсгэх хоёр чиглэлт ажлыг төгсгөлөг төлөвт автоматд үндэслэн монгол хэлний үг зүйг загварчлан шинэ хэрэглүүр бүтээсэн. Энэхүү зохиосон хэрэглүүрээ ашиглан монгол хэлний үгсийг хэрхэн бүтээж, хэрхэн таньж бүтээврүүдэд задалж байгаа үр дүнгээс харахад үг хэрэгслийг ашиглан монгол хэлний үг зүйд шинжилгээ хийх боломжтой бөгөөд үр дүнтэй болох нь батлагдлаа. Монгол хэлний үг зүйг загварчилж өгснөөр бид компьютер хэл шинжлэлийн бусад салбаруудыг хөгжүүлэх боломжтой гэж үзэж байна.

Бидний энэ судалгааны ажил нь монгол хэлний хэл шинжээчид, хэл шинжлэлийн салбарын оюутнуудад тус болох төдийгүй цаашид хийх шаардлагатай монгол хэлний өгүүлбэр зүйн задлагч, монгол бичгийн хөрвүүлэгч зэрэг программ хангамжийг хийж гүйцэтгэх үндэс, суурь нь болно гэдэгт найдаж байна. Бидний төгсгөлөг автоматыг машин сургалтын зарим алгоритмтай хамт ашиглан боловсруулсан үг зүйн энэ загварын энэхүү судалгаа нь математик, компьютерын ухаан, хэл шинжлэлийн салбарыг холбосноороо онцлог юм.

Ажлын явцад бидний боловсруулсан энэхүү хэрэгсэл нь монгол хэлний кирилл ба монгол бичгийн үг зүйн загварчлалд хэрэглэж болох нь тодорхой болсон ба энэхүү үг бүтээж, задалж байгаагаа дараагийн шатны судалгаандаа ашиглах бүрэн боломж нээгдлээ. Үг хувилгах автоматын дагуу монгол хэлний үг зүйн шинжилгээг амжилттай хийж гүйцэтгэсэн ба үгийн утга таних сургалтын санг хангалттай хэмжээнд бүрдүүлж өгсөн нөхцөлд бидний сонгосон аргаар монгол үгийн утга танихад боломжтой үр дүнг үзүүлэхээр байна. Бидний боловсруулсан кирилл болон монгол бичгийн үг зүйн загварчлалын программ хангамжийн тухай тайлангийн 2-р бүлгээс дэлгэрүүлэн уншина уу.

Бичвэрийн алдааг илрүүлж засах алгоритмууд, алдааны төрлүүд, үгийн алдааг илрүүлэх, засах талаар судалж түгээмэл хэрэглэгддэг алдаа шалгуурын алгоритм болох Levenshtein, N-Gram аргыг аргуудыг ашиглан кирилл болон монгол бичгийн бичвэрээс алдааг илрүүлж, засах туршилтыг хийлээ. Үгийн алдааг бичиглэлийн алдаа (Non-Word Error) буюу утгагүй алдаа, үг сонголтын алдаа (Real-Word Error) буюу утгатай алдаа гэж хоёр хуваан авч үзсэн. Үгийн бичиглэлийн алдааг олохдоо Левенштэйний алгоритм буюу хоёр тэмдэгтийг ойролцоолох (тэмдэгтийн зөрүүг олдог) алгоритмд тулгуурласан бол N-Gram аргыг үгийн утгын алдааг олоход ашиглаж холбогдох туршилтыг хийлээ. N-Gram аргыг Back-off soomthing, Sparse Matrix, MLE (Maximum Likelihood Estimation), MED (Minimum Edict Distance) зэрэг аргуудын хамт ашиглалаа.

Туршилт үр дүнгээс харахад Левенштэйний алгоритмыг дангаар нь ашиглахад зөвхөн тухайн үгийн алдаатай эсхийг шалгаж байгаа бол N-gram аргыг ашигласнаар утгын алдаа буюу үг сонголтын алдааг санал болгож байна. ӨМИСургуулын хийсэн алдаа шалгуур дээр шалгаж үзэхэд алдаатай үгийг засаж мөн утгын алдааг тодорхой хэмжээнд илрүүлж байна.

Үгийн алдаа шалгуурын программ хангамжийг хөгжүүлж, тодорхой туршилт хийхийн тулд нөхцөлийн дарааллийн сан бүрдүүлэх, эхэд анализ хийх, бичвэрийн сан бүрдүүлэх зэрэг холбогдох зарим программ хангамжийг хөгжүүлж ашигласан. Программ хангамжийн үр дүн нь ямар текст оруулж байгаагаас болон үгийн санд хэдэн үг байгаасаа шууд хамааралтай байгаа учраас программын үр дүнг сайжруулахын тулд өгөгдлийн сангаа маш сайн өргөжүүлэх шаардлагатай байна. Бидний боловсруулсан кирилл болон монгол бичгийн бичвэрийн алдаа илрүүлж засах арга зүй, туршилт түүний үр дүн, зарим нэгэн программ хангамжийн тухай тайлангийн 3-р бүлэгт дэлгэрэнгүй өгүүлсэн болно.

Кирилл болон монгол бичгийн хооронд харилцан хөрвүүлэх алгоритмыг монгол хэлний дүрэмд, статик өгөгдөлд суурилсан болон машин сургалтын аргыг ашиглан боловсруулан туршлаа.

Монгол хэлний хоёр бичгийн хөрвүүлгийн хувьд үүсмэл хэлэнд голчлон хэрэглэгддэг дүрмэнд суурилсан аргыг хэрэглэх нь илүү оновчтой нь бидний судалгааны ажлаас харагдсан бөгөөд кирилл болон монгол бичиг нь өгүүлбэр зүйн бүтцийн хувьд адилхан тул монгол хэлний эдгээр хоёр бичгийн хувьд үг зүйг нь загварчлахад хангалттай байсан. Бичвэр боловсруулах ажлын үр дүн нь тэмдэгтийн кодлолтоос (латин, кирилл, монгол бичиг гэх мэт) хамааралгүй бөгөөд гол нь үгийн

сангийн файлаа хэр хангалттай бүрдүүлж, зөв зүйтэй ангилав, дүрмээ хэр зөв тодорхойлж загварчлав гэдгээс шууд хамаарч байна. Тиймээс монгол хэлийг кирилл бичиг, уламжлалт монгол бичиг, латин үсгийн алинаар нь кодолж, түлхүүрдэж байгаагаас үл хамааран бидний боловсруулсан үг зүйн загварчлалын хэрэглүүрийг ашиглаж болж байна.

Энэхүү хоёр бичиг хооронд хөрвүүлэх гэдэг нь нэг бичгээр өгөгдсөн үгийг бүтцээр нь задалж, үгийн үндэс болон нөхцөлүүдийг олно. Ингээд өгөгдлийн сангаа ашиглаж үгийн үндэс ба нөхцөлүүдийн нөгөө бичгээрх хэлбэрийг тодорхойлно. Эцэст нь олж тодорхойлсон үндсийг өгөгдсөн нөхцөлүүдээр хувилгах замаар нөгөө бичигтээ хөрвүүлнэ. Хөрвүүлэх ерөнхий зарчмыг болон үр дүнг 4-р бүлэгт танилцуулав. Кирилл бичгээс монгол бичигт хөрвүүлэх үед кирилл үг нь олон хэлбэрээр бүтцээр задрах болон үгийн утгаас хамаарч монгол бичгээр ялгаатай бичигдэх боломжтой байдаг. Иймээс үгийн утга таних асуудал чухлаар тавигдана. Монгол бичгээс кирилл бичигт хөрвүүлэх үед монгол бичгээр бичигдсэн оноосон нэрийг ялгаж таних, мөн үгийн утга таних асуудал чухлаар тавигдаж байна. Туршилт хийсэн бичвэрийн зөв бичих дүрмийн алдаанаас болж хөрвүүлээгүй буюу буруу хөрвүүлсэн үг байгаа учраас хөрвүүлэхийн өмнө хоёр бичгийн бичвэрийн алдааг шалган зөв бичих дүрмийн дагуу нэг бүрчлэн засаж дахин хөрвүүлэлт хийх нь зүйтэй.

Их хэмжээний өгөгдөл дээр дараалалтай холбоотой асуудлыг шийдэхэд RNN (Recurrent Neural Network) нь хамгийн тохиромжтой машин сургалтын арга бөгөөд яриа таних, эх хэлний боловсруулалт (NLP), хугацаанаас хамаарсан таамаглал дэвшүүлэх зэрэг олон төрлийн хэрэглээтэй. RNN архитектурын тусгай арга болох Sequence to Sequence (seq2seq) загварыг машин орчуулга, асуултанд хариулах, чатбот үүсгэх, бичвэрийг хураангуй болгох гэх мэт хэлний нарийн төвөгтэй асуудлыг шийдвэрлэхэд ихэвчлэн ашигладаг. Бид ч гэсэн энэ аргыг ашиглан монгол хэлний хоёр бичгийн бичвэр хооронд хөрвүүлэх туршилтыг хийж үзсэн. Ингэхдээ I бүлэгт дурдсан кирилл-монгол бичгийн 90.000 гаруй холбоо үгийн болон харгалзсан 50.000 өгүүлбэр бүхий сангаа ашигласан бөгөөд энэ сангаас сургахад хугацаа их авч байсан тул эхний ээлжинд 2000 холбоо үг, өгүүлбэрийг ашиглан RNN-ны (learning rate 0.05, 128 нейрон сүлжээ) seq2seq загвараар сурган туршив. Ингэхдээ машин сургалт ашиглан хөрвүүлэг хийх аргыг ИХ өгөгдөл, БАГА өгөгдөл дээр олон удаа туршиж хамгийн үр дүнтэй гэсэн 4 туршилтын үр дүнг тайланд тусган харуулсан.

Дүгнэлт

Энэ сэдэвт ажлын хүрээнд төлөвлөгдсөн ажлууд амжилттай хийгдэж дууссан бөгөөд тус тусын дүгнэлт сэдэв болгонд хийгдсэн байгаа. Сэдэвт ажлыг гүйцэтгэснээр монгол хэлний кирилл болон монгол бичгийн бичвэрийн цахим сан бий болж монгол хэлний үг зүйн загварчлалын хэрэгслийг шинээр бий болгон бүтээсэн. Мөн машин сургалтын зарим аргуудыг дүрэмд суурилсан болон статик аргуудтай хослуулан бичвэрийн алдаа шалгах болон бичвэр хооронд хөрвүүлэх алгоритмуудыг боловсруулан түүнийхээ дагуу холбогдох программ хангамжуудыг бичин туршилт хийсэн.

Эндээс голлон дүгнэвэл монгол хэлийг цахим орчинд боловсруулах онол арга зүй боловсруулагдаж, гарсан үр дүн дээр бичвэр хураангуйлах, асуултад хариулах, машин орчуулга, ярианаас бичвэрт, бичвэрээс ярианд хөрвүүлэг хийх зэрэг системүүдийг хөгжүүлэх боломж бүрдсэн бөгөөд энэхүү ажлыг цааш үргэлжлүүлэн хэрэгжүүлэх нь зүйтэй болно. Ажлын үр дүнгээр эрдэм шинжилгээний өгүүлэл 5 хэвлэгдэж, 1 өгүүлэл хяналтын шатанд байгаа бөгөөд олон улсын болон дотоодын хурал семинарт 12 илтгэл тавьж хэлэлцүүлсэн.

Хэвлэгдсэн бүтээлүүдийг дор жагсаартаар орууллаа. Тайлангийн эцэст бүх ашигласан материалыг дэлгэрэнгүй тоочсон болно.

Судалгааны хүрээнд хэвлэгдсэн эрдэм шинжилгээний өгүүлэл

Дотоод:

1. “Монгол хэлний нөхцөлийн сан бүрдүүлэх” М.Хүрэлхүү, Д.Уганбаатар, И.Бямбасүрэн, ШУА, ФТХ-ийн эрдэм шинжилгээний бүтээл №45. х.141-149, 2018
2. “Монгол бичгийн бичвэрээс алдааг илрүүлж засах нь”, М.Хүрэлхүү, Д.Уганбаатар, И.Бямбасүрэн, М.Мархамет, МТТХ-ийн эрдэм шинжилгээний бүтээл №1. х.29-37, 2019, ISSN: 2708-0242
3. “Компьютерын үг зүйд зориулсан монгол хэлний цахим хөмрөгийн бүтээврийн ангилал” Т.Пүрэвсүрэн, “Хэрэглээний хэл шинжлэл” сэтгүүл, №4 (19). х.128-142, Уб, 2019.

Гадаад:

1. “Mongolian Language Morphology and Its Database Structure”, Uuganbaatar Dulamragchaa, Sodoo Chadraabal, Byambasuren Ivanov, Munkhbayar Baatarkhuu, 2017 International Conference on Green Informatics (ICGI), pp. 282-285, 2017, http://doi.ieeecomputersociety.org/10.1109/ICGI.2017.56 (doi:10.1109/ICGI.2017.56)
2. “Recognition of traditional Mongolian script using primitives and template matching methods”, I.Byambasuren, D.Uuganbaatar, M.Markhamet, O.Otgonnaran, International Journal of Scientific & Engineering Research Volume 10, Issue 2, February-2019, ISSN: 2229-5518, (DOI: 10.14299/ijser.2019.02.02)

Түлхүүр үг

Компьютер хэл шинжлэл, кирилл, монгол бичиг, программ хангамж, машин сургалт, хөмрөг

I. ЦАХИМ ХЭЛ ШИНЖЛЭЛД ЗОРИУЛСАН ӨГӨГДЛИЙН САНГ БҮРДҮҮЛЭХ НЬ

Компьютер хэл шинжлэлийн судалгаанд зориулан тусгай зориулалтын төрөл бүрийн өгөгдлийн сангуудыг зохион байгуулдаг. Ажлын явцаас шалтгаалан төрөл зүйл, дотоод бүтэц, зохион байгуулалт, бүрдүүлэх аргатай холбоотой асуудлуудыг тусгайлан судалдаг Corpus Linguistics гэсэн салбар ч бий болсон.

Цахим хөмрөг бол монгол хэлний үгийн сангийн цахим орчин дахь илэрхийлэл болно. Үгийн сан бол тухайн ард түмний ахуй амьдрал, ёс заншил, шашин шүтлэг, хэл соёл, нийгэм улс төр, гадаад харилцаа зэргийг шууд тусгаж, нийгмийн хөгжил хувьслыг дагалдан зарим хэсэг нь хуучирч мартагдах аястай байхад, зарим нь эргээд тэнхрэн хэрэглээнд идэвхжин орж ирэх, шинэ соёл, шинэ юмыг даган бусад хэлнээс үг орж ирэх, шинэ үг бүтээх, шинэ утга, шилжсэн утга бий болох зэрэг олон аргаар баяжиж, өөрчлөгдөн хувьсаж байдаг жамтай.

Цахим хөмрөг байгуулж ирсэн байдлыг харвал, орчин үеийн, дунд үеийн гэхчлэн цаг хугацааных нь үүднээс үечлэх, идэвхтэй, идэвхгүйг харгалзан хэрэглээний үүднээс хандах, тогтмол үг хэллэгийн, гадаад, ижил, эсрэг, ойролцоо утгатай үг хэллэгийн гэх мэтээр тусгай зорилгодоо нийцүүлэн сан бүрдүүлж иржээ.

Бүрдүүлсэн энэхүү өгөгдлийн сан нь компьютер хэл шинжлэлийн талаар цаашид бидний хийх олон ажлын суурь болно. Сангаа ашиглан хамгийн эхэнд монгол хэл, монгол бичгийн үг зүйн болон хооронд нь хөрвүүлэх системийн судалгаа хийж гүйцэтгэх болно.

Хөрвүүлэх программын зорилгыг хэрэглэгч аль ч цаг үеийн, ямар ч төрлийн бичвэрийг хөрвүүлэх хэрэгцээ, сонирхолтой байж болно, түүнийг аль болох бүрэн хөрвүүлэх хэрэгтэй гэж тодорхойлж байна. Энэ нь бидний программын хувьд ажиллах объект тодорхойгүй, хэрэглэгчээс шууд хамааралтай байна гэсэн үг.

Ингэхийн тулд өгөгдлийн сангийн цар хүрээ нь цаг хугацааны аль ч цагт хамаарах, албан бичгийн, уран зохиолын, эрдэм шинжилгээний гэх мэт найруулгын ямар ч түвшний эхийг аль болох бүрэн гүйцэд хөрвүүлэхэд хүрэлцэхүйц хэмжээнд байх ёстой гэсэн шаардлагатай тулгарна. Өөрөөр хэлбэл “*ерөнхий сан”* байна гэсэн үг.

1.1 Өгөгдлийн санг бүрдүүлэх арга, хэрэглэгдэхүүн, хэмжээ

Энд анхаарвал зохих зүйл бол одоогийн түвшинд бидний авч үзэж байгаа хангалттай хэмжээний өгөгдлийн чанар нь “зөв” байх зарчмын дотор хэрэгжинэ. Эхний ээлжинд монгол бичгийн хэлний хэм хэмжээнд нийцсэн “зөв хөмрөг”, түүнийг “зөв хувиргах дүрэм”-ийн цогцыг бүтээнэ гэж хэлж болох юм. Харин цаашдаа хөрвүүлэх программын дараагийн хувилбар, ирээдүйд бий болгох шаардлагатай байгаа зөв бичлэг шалгуур, хэлзүйн шалгуур, орчуулга зэрэг программ боловсруулах үед үгийн үндсийг буруу бичдэг, буруу хувилгадаг зэргийг аль болох арилгах боломжийг хангахыг хичээлээ.

Ер нь ч компьютер хэл шинжлэлийн аливаа салбарын судалгаа, түүнээс гаргасан программ эхэн үедээ “цэвэр” өгөгдлийн хүрээнд ажилладаг. Аажмаар хэрэглээнээс ажиглалтын туршлага хуримтлуулж, түүндээ суурилан өгөгдлийн алдаатай хэсэгтэй

ажиллах буюу булингарыг тунгаахад чиглэсэн технологийн болоод хэлний боловсруулалт хийж, дараагийнхаа сайжруулсан хувилбарыг гаргадаг билээ. Жишээ нь, 1960-аад оноос хөгжиж эхэлсэн асуултад хариулах салбарын “Text Retrieval Conference” (TREC) гэхэд ердөө 3 жилийн өмнөөс л блогийн өгөгдлийг оруулж ирсэн билээ. Блогийн өгөгдөл бол жинхэнэ бодит амьдрал дээр хүн шууд л бичээд орхидог, нухацтай хянах талаар санаа тавьдаггүй, зөв буруу хутгалдсан ярвигтай хөмрөг юм.

Иймээс “*цэвэр өгөгдөл*”-ийн гадуур үлдэж байгаа хэсэг бол зохиогч өөрийн үзэл баримтлалаа илэрхийлж зориудаар бичсэн зарим бичлэг, санамсаргүй буюу бичигчийн боловсролоос шалтгаалсан буруу бичлэг, сүүл үед маш их элбэгшиж байгаа гадаад үгийн бичлэг зэрэг болно. Энэ гурван зүйлийг товч тодруулъя.

Хөмрөг бүрдүүлсэн арга

Эхний шатны харьцангуй зөв, цэвэр өгөгдлийг авахын тулд:

- Хамгийн эхэнд толь бичгийг хуудас бүрээр эргүүлж толгой үгийг цахим тооцоолуурт шивж оруулав.
- Дараагаар нь, кирилл үсгээрх толгой үгэнд харгалзах монгол бичгээрх бичлэгийг харгуулав.
- Гуравт харгалзах утгыг нэмж эхэлсэн.
- Эцэст нь толгой үгээ оруулсны дараагаар монгол хэлний зүйн хувиллын загварт тулгуурлан үг бүрт тохирох тэмдэглэгээ буюу хувиллын кодыг хадсан.

Энэ үе бүхэлдээ гар аргаар бий болдог тул тус салбарт ажиллаж буй хэлний мэргэжилтнээс их хэмжээний өгөгдөл доторх нэгж бүрийг бодож, няхуур хандахыг шаарддаг онцлог бийг дурдах хэрэгтэй болов уу.

Хэрвээ хэлний мэдлэгийг “барилга” хэмээн зүйрлэвэл, цахимын бус судалгаанд тэр барилга юунаас бүтдэг, ямар загвараар барьсан, аль давхарт нь юу юу байдгийг түлхүү анхаарсан ерөнхий тогтолцоог нь мэдэж, тэр тогтолцоондоо тохирсон загвар жишээтэй харьцаж байдаг бол компьютер хэл шинжлэлд ажиллахын тулд тэрхүү барилгынхаа өрөө бүрийн хана, тааз, цонх гээд бүх эд ангитай нүүр тулах болдог. Тиймээс ч өвөр монголд “чилээн барах арга” гэдэг юм байна.

- Дараагийн шатанд бэлэн байгаа бичвэрүүдийн үгийн хэлхээг гаргаж, тэндээс харьцуулах замаар баяжуулах аргаар явсан. Хэлхээ гаргах зэрэгт программ ашиглах боломжтой, өмнөхөөс бага цаг зарах ч бас л амаргүй, бүрэн сайн шийдэл бус. Товчоор, хагас автомат гэж болно.
- Хамгийн боловсронгуй арга бол нэгэнт буй болгосон тэмдэглэгээ бүхий хөмрөгийг ашиглан үгийг хувилалтай нь үүсгэж, түүнийгээ интернэт, бодит хэрэглээн дээрх эх бичвэрүүдтэй тулгах замаар бүрэн автомат аргаар шийдэх боломжтой. Боловсруулалтын дүнд гарсан хөмрөгт байхгүй үгсийг шалгах нь хэлний мэргэжилтний хянах үүрэг. Энэ нь нэг талаас автоматаар гаргасан өгөгдөлтэй ажиллах, нөгөө талаас программын загвар, зохиомж, хөмрөгтэй ажилласан дадлага туршлагатай болсон байдгаас хамааран өмнөх хоёр түвшинтэй харьцуулахад маш бага цаг зарна.

1.2 Үндсэн өгөгдлийн сангийн бүтэц

Өгөгдлийн сан нь бидний бүх ажлын суурь болох учраас цаашид ашиглах сангаа судалгааныхаа зорилгоос хамаарч өөрийн гэсэн онцлогтойгоор бүрдүүлэх, бүтэцийг зөв тогтоон зохион байгуулах нь нэн тэргүүнд анхаарах ёстой асуудал болдог. Өгөгдлийн сангаа байгуулсан арга, зарчмын талаар товчхон дурдъя.

Үндсэн санг бүрдүүлэхдээ 2007 онд Шинжлэх Ухааны Академийн Хэл Зохиолын Хүрээлэнгээс эрхлэн гаргасан “Монгол хэлний дэлгэрэнгүй тайлбар толь” 5 боть зохиолыг ашиглалаа.

Нийт өгөгдлийн санг бүрдүүлсэн явц:

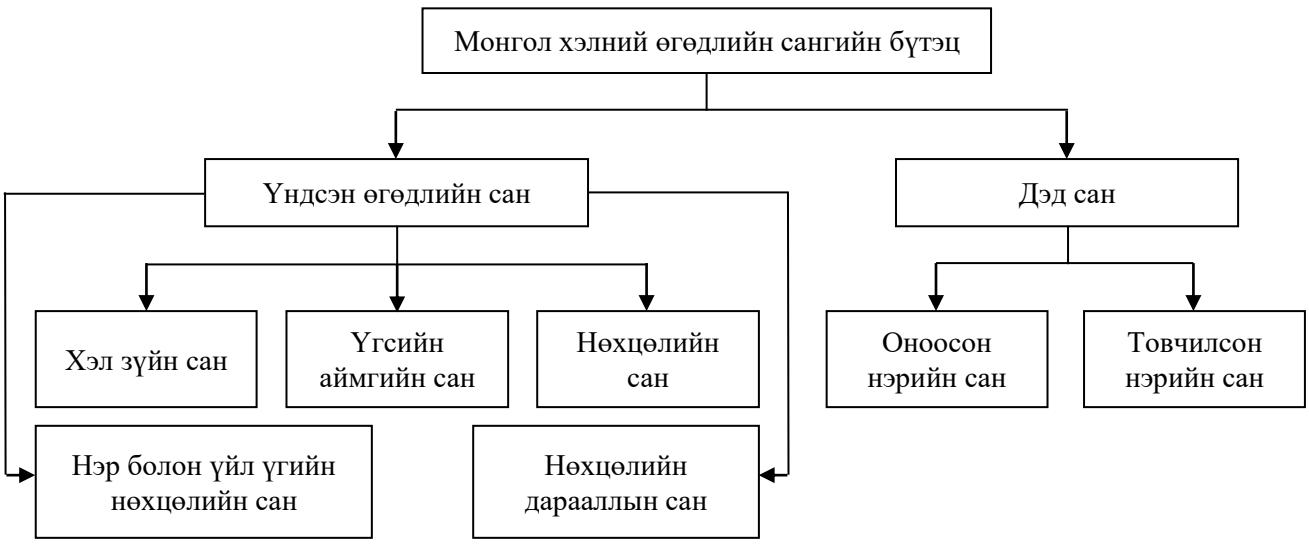
- Нэгд, толь бичгийн толгой үгээр үндсэн талбар болгон оруулав.
- Хоёрт, кирилл толгой үгэнд харгалзах монгол бичгээрх харгалзах бичлэгийг оруулсан.
- Гуравт, толгой үгээ оруулсны дараагаар түүнд харгалзах тайлбар болон зарим нэгэн холбоо үгсийн санг бүрдүүлсэн болно.
- Толгой үгэндээ анхдагч түлхүүр оноосон. Энэ нь сангуудын хоорондох хамаарлыг тогтоож өгнө.
- Толгой үгэнд харгалзах үгсийн аймгийг тогтоож хадав.
- Монгол хэлний үгийн хувиллын дүрэмд тулгуурлан үг бүрт тохирох тэмдэглэгээг хийсэн.

Үндсэн өгөгдлийн санд дараах 3 төрлийн санг хамааруулан ойлгож болох юм.

- Анхдагч түлхүүр, кирилл, монгол толгойн үг болон тайлбар бүхий анхдагч өгөгдлийн сан
- Үгсийн аймаг бүхий сан
- Хэл зүйн хувиллыг заах код бүхий сан хэл зүйн сан

Эдгээр нь ерөнхийдөө гар аргаар бий болдог тул их хэмжээний өгөгдөл дотор ажиллан нэлээд цаг хугацаа шаардсан онцлог бүхий ажил юм.

Одоо сан бүрийн бүтэц хоорондын холбоог дараах зурагт харуулав.



Зураг 1. Өгөгдлийн сангийн бүтэц

Ингэснээрээ хоёр түвшинт үг зүйн аргыг ашиглан монгол хэлний үгсийг хэрхэн бүтээж, хэрхэн таньж бүтээврүүдэд задалж байгааг харуулах бүрэн боломжийг бүрдүүлсэн.

Үгсийн сан

Үгийн санг хэрэглээгээр нь ерөнхий ба тусгай зориулалтын гэж ангилж болно. Ерөнхий зориулалтын үгийн сан нь нийтийн хэрэгцээнд зориулагдсан, тусгай зориулалтын үгийн сан нь мэргэжлийн хүмүүсийн хэрэгцээнд зориулагдсан байдаг [1]. Мөн монгол хэлний үгийн сангийн үгийг идэвхи хэрэглээгээр нь идэвхитэй ба идэвхигүй гэж ангилж болно [1]. Санд одоогоор үгийн 50842 үндэс оруулсан ба үүнээс нэр үг 27941, үйл үг 22739 байна.

Үгийн үндсийг бүлэглэх нь

Монгол хэлний үгийн хувилал нь уг үгийн төгсгөлөөс хамаардаг гэдгийг Ц.Дамдинсүрэн, Б.Осор нар тогтоож монгол хэлний толь хийхэд энэ чанарыг ашигласан байна [2]. Тэд "Нэр, үйл үгсийг хэдэн хэлбэрт хувааж, хэлбэр бүрийг тоогоор дугаарлан, тэр дугаар нь тухайн үг яаж нөхцөл авч хувирахыг хавсралт жагсаалтаас олж үзэж болно. Ингэж нэр ба үйл үтийн олон хэлбэрийг бүрэн жагсаахгүй, тэр олон хувилбарыг эцсийн жагсаалтаас үзэхээр хийсэн нь уг толийг авсаархан болгоход тус боллоо" гэжээ. Өөрөөр хэлбэл нэр, үйл үгийн үндсийг төгсгөлөөс нь хамааруулж дугаарлажээ.

Энэ нь үнэндээ үгсийг төгсгөлөөр нь ижилсүүлэн эгшиг гээгдэх, үсэг жийрэглэх гэх мэт зөв бичгийн дүрмүүдийг баримтлан бүлэг болгон хуваах санаа байсан юм. Хэрэв нэг ижил дугаартай үгсийг нэг бүлэг болгож авбал тэр бүлгийн бүх үгийг нэг ижил дүрмээр хувилдаг гэж үзжээ.

Иймээс ижил төгсгөлтэй үгийн үндсүүд ижил дүрмээр хувилна. Иймээс үгийг төгсгөлөөр нь бүлэглэж болно.

Энэ санааг ашиглан “үгсийн хэлбэр” гэдгийг үгийн бүлэг болгон нэрлэж, ижил төгсгөлтэй бөгөөд аливаа нөхцөлийг залгах үед нэг ижил дүрмээр хувилдаг үгсийг нэг бүлэг болгож кодлов. Ингэж ижнл төгсгөлтэй үгийн үндсүүд нэг бүлэгт хамаарна. Энэ нь судалгаагаар кирилл бичигт нэр үгийн үндсийг 32, үйл үгийн үндсийг 16 бүлэгт хувааж, харин монгол бичигт нэр үгийг 6, үйл үгийг мөн 6 бүлэгт хувааж бүлэглэсэн байна [6].

Үгийн бүлгийн код

Нэр үг болон үйл үгийн үндсүүд нь ижил төгсгөлтэй боловч ялгаатай нөхцөлөөр хувилдаг тул нэр ба үйл үгийн бүлгийг ялгаатай үзэх шаардлагатай. Мөн Ц.Дамдинсүрэн, Б.Осор нарын үг бүлэглэсэн байдлыг харахад үгийн төгсгөлөөр ялгахад үгийн эгшиг оролцож, түүнд залгах нөхцөлийн эгшгийг тодорхойлж байна. Иймээс кирилл үгийн үндэс бүрийн хувьд:

- үгийн аймаг (нэр үг эсвэл үйл үг)
- бүлгийн дугаар
- үгийн эгшиг оролцсон код үүсгэж болно.

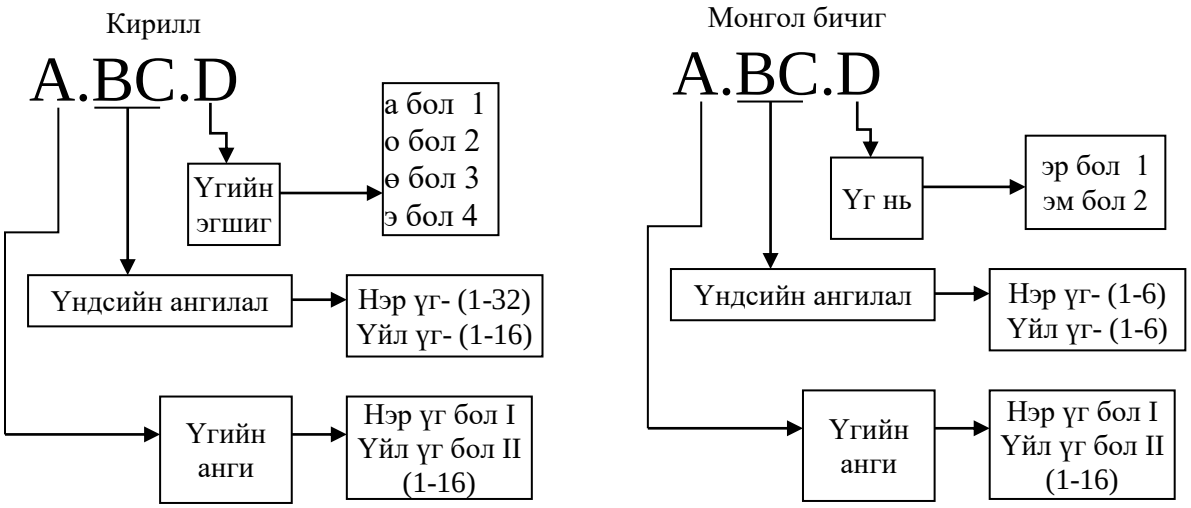
Харин үүнтэй төстэйгээр монгол бичгийн үгийн үндэс бүрийн хувьд:

- үгийн аймаг (нэр үг эсвэл үйл үг) бүлгийн дугаар
- үгийн хүйс (эр үг эсвэл эм үг) оролцсон код үүсгэж болно.

Эндээс үзвэл, үгийн үндэс бүр нь тодорхой тооны атрибут (шинж чанар)-тай байх бөгөөд нөхцөл залгах үед үгийг хувилах дүрмийг эдгээр атрибутаар нь тодорхойлно. Энэхүү шинж чанарыг харуулж Ш.Чоймаа, Э.Мөнх-Учрал нарын эрдэмтэд “Монгол хэлний хэл зүйн” толь бүтээснийг өгөгдлийн сандаа шууд авч ашиглалаа. Үгийн кодын зохиомжийг зураг 2-т харуулав.

Нэг үгэнд кирилл болон монгол бичгийн гэсэн 2 код харгалзана. Энэ үгийн код нь өмнө өгүүлсэн хэл зүйн санд хадгалагдана. “**Үгийн код**” гэдгийг үгийн үндсийн тухай тодорхойлолт, үндсэнд хамгийн эхэнд залгагдах нөхцал, бүтээвэрийн зохистой хэлбэрийн олонлог, зөв бичих зүйн дүрмээр үг, түүний ард шууд тохиолдох эхний нөхцөл залгавар хоёрыг холбон бичих дүрмийн цогц мэдээлэл хэмээн тодорхойлж болно.

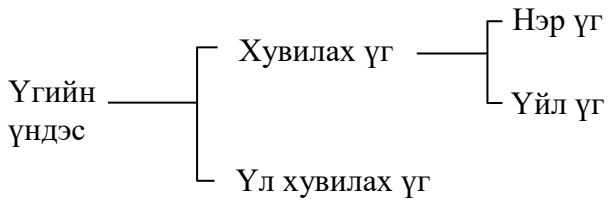
Өөрөөр хэлбэл, үүгээр дамжуулан автоматын дагуух эхний шилжүүлгийг гүйцэтгэх хоёр бичгээрх өгөгдөл, дүрмийг тооцоолж буй юм.



Зураг 2. Үгийн кодын зохиомж

1. **A** - Тухайн үгийн үндэс нэр эсвэл үйл болохыг тодорхойлно.
2. **BC** - Үгийг хувилгахад зөв бичих дүрмээр нь бүлэглэсэн дугаар. Үүнд нэг талаас тухайн үгийн дүрмийн онцлог, нөгөө талаас түүнтэй уялдсан залгаврын олонлог харгалзана.
3. **D** - Тухайн үндсийн агуулсан эгшгээс шалтгаалан, залгагдах залгавын эгшгийг тодорхойлно.

Өгөгдлийн санд хадгалагдсан үгсийг дараах байдлаар тодорхойлов. Монгол хэлний үгийн үндсийг хувилах нөхцөлийн шинжид нь тулгуурлан нэр үг, үйл үг, үл хувилах үг гэж 3 ерөнхий аймаг болгон хуваах боломжтой [3][4]. Үүнийг Зураг 3-д харуулав.



Зураг 3. Сан дахь үгийн ангилал

Эхийн доторх бусад үгсийн аймгийн, жишээлбэл тэмдэг нэр нь жинхэнэ нэрийн үүргээр орсон гэх мэт хувилсан тохиолдлууд нь үүнд бүрэн шингэж чадах учраас энэ ангилал бидний зорилгод нийцэж буй. Хувилах шинжтэй нэгжид кодыг хадан дараа, дараагийн шатны боловсруулалтад орох ба үл хувилах үг кодгүй, зөвхөн харгуулсан бичлэгээр хөрвөх болон шаардлагатай тохиолдолд бичлэг онтусгах гэсэн нэгээс хоёр үйлдлээр дамжиж программын үр дүн болон гарна.

Кирилл үсгийн дүрмийг бүлгийн кодоор илэрхийлэх нь

Кирилл үсгийн дүрэм анх 1942 онд хэвлэгдэж [Дамдинсүрэн, 1942], 1946 онд дахин засварлан гаргасаныг өмнөх бүлэгтээ дурдсан.

Энэхүү дүрэм нь орчин цагийн монгол хэлний халх аялгууны байдлыг харгалзсан боловч монгол хэлний авианы байрлалын хуулийг бүрэн тусгаагүйн улмаас олон зохиомол дүрэмтэй, тэдгээр нь үгийн язгуур, үндэс, дагавар, нөхцөлийн бүтцийг бараг бүх тохиолдолд эвддэг. Бид хүчинтэй буй кирилл үсгийн дүрмийг мөрдсөн бөгөөд үүний дагууд боловсруулахад олон зүйл бэрхшээл тохиолдож байна.

Нэр үгийн бүлэг

Кирилл үсгийн дүрмээр нэрийн хувилах бүлэг 32. Эдгээр бүлэг нь эгшгийн зохицлоос хамааран дотроо 1-4 хүртэл хуваагдана. Дашрамд дурдахад, тухайн үг олон тоонд хэрхэн хувирах нь утгаас шууд хамаардаг. Тэгэхээр нэр үг бүрт ямар олон тооны залгавар авч болох хийгээд тэрхүү залгавар нь үндсийн ямар хэлбэрт залгагдахыг тэмдэглэж өгсөн тухай өмнө дурдсан билээ.

32 бүлгийг тус бүрээр товч нь задлан тайлбарлавал [6],

1-р бүлэгт эгшигт гийгүүлэгчээр төгссөн, балархай эгшиггүй, үндсэнд “*н*” гардаггүй үгс багтана. Нэрийн нөхцөлийг эгшгийн зохицлыг харгалзан шууд залгана.

Жишээ: 1.1 *санал, цуврал* 1.2 *ном, бодрол*
1.3 *хөл, өмсгөл* 1.4 *бэр, дэглэм*
саналууд, саналын, саналд, саналыг, саналаас, саналаар, саналтай, санал руу, саналаа,

2-р бүлэгт “*н*” – ээр төгссөн, балархай эгшиггүй, үндсэнд нь “*н*” гардаггүй үгс багтана. Гол онцлог нь харьяалахын тийн ялгалын “*-ы, ий*” гэсэн нөхцөлийн хэлбэр авна.

3-р бүлэгт заримдаг гийгүүлэгчээр төгссөн, балархай эгшиггүй, үндсэнд “н” гардаггүй үгс хамаарна. Энэ бүлгийн үгэнд өгөх оршихын тийн ялгалын “-д” хэлбэрийг залгахад зохих эгшгийг жийрэглэнэ.

4-р бүлэгт “*ж, ч, ш*”-ээр төгссөн, балархай эгшиггүй, үндсэнд “*н*” гардаггүй үгс орно. Онцлог нь үгийн эр, эмийг үл харгалзан харьяалах, заахын тийн ялгалд “*-ийн, -ийг*” гэсэн хэлбэрийг залгаж, өгөх оршихын тийн ялгалд “*и*” эгшиг жийрэглэнэ.

5-р бүлэгт балархай эгшиггүй, үндсэнд “*н*” гардаггүй үгс буюу монгол бичигт дүйцэх нь хатуу дэвсгэрээр төгссөн үгс багтана. Өгөх оршихын тийн ялгалын “*т*” хэлбэрийг шууд залгадгаараа ялгарна.

6-р бүлэгт “*г*” гийгүүлэгчээр төгссөн, ялгах эгшиггүй, балархай эгшиггүй үгс хамаарна. Онцлог нь эр үг болохыг нь үл харгалзан харьяалах, заахын тийн ялгалд “*-ийн, -ийг*” гэсэн хэлбэрийг залгаж, өгөх оршихын тийн ялгалд “*-т*” нөхцөл залгана.

7-р бүлэгт туслах эгшгээр төгссөн үгс хамаатай. Дүрмийн онцлог нь харьяалах, заахын тийн ялгалаас бусад урт эгшгээр эхэлсэн нөхцөл залгахад нэг эгшгийг гээж бичнэ. Туслах эгшгийн тооноос хамаарч 7-р бүлэг 3 хувилбартай.

8-р бүлэгт богино “и”-ээр төгссөн үгс орно. Бусад бүлгээс ялгарах онцлог нь урт эгшгээр эхэлсэн бүх залгавар нөхцөлийг залгахад нөхцөлийнх нь эхний эгшгийг гээж бичих буюу программд ойлгуулснаар нөхцөлийн нэг эгшигтэй хувилбар (*-ар, эс, йн* г.м)-ыг залгана.

9-р бүлэгт үгийн үндсэнд “*н*” гардаг, балархай эгшиггүй үгс багтана. Заах, үйлдэх, хамтрах, чиглэхийн тийн ялгал, хамаатуулахын нөхцөлийг шууд, нэрийн бусад нөхцөлийг залгахад “*н*” гарч, өгөх оршихын тийн ялгалд зохих эгшгийг бас жийрэглэнэ.

10-р бүлэгт “*к*” үсгийг хэлний “*г*” үсгийн дүрмээр үзэх [Дамдинсүрэн, 1983, х.422] дүрмийн дагуу гарах бүлэг бөгөөд ойролцоо шинжтэй “I.6.1, I.6.2”-оос тогтворгүй “*н*”гарахаараа ялгарна. Жишээ: 10.1 *банк, танк*

11-р бүлэгт “*ж, ч, ш*”-ээр төгссөн, үндсэнд “*н*” гардаг, балархай эгшиггүй үгс орно. 4-р бүлгээс “*н*” гардгаараа ялгаатай.

12-р бүлэгт урт эгшгээр төгссөн үгс багтана. Олон тоо, харьяалах, өгөх орших, гарахын тийн ялгалд нөхцөлийн “*н*” бүхий хэлбэрийг, заахын тийн ялгалд “*г*” хэлбэрийг, үйлдэх, эзэнд хамаатуулах нөхцөлд “*г*” жийрэглэнэ.

13-р бүлэгт богино “*и*”-ээр төгссөн, нууц “*н*” бүхий үгс хамаатай. 8-р бүлгээс үндсэнд “*н*” гардгаараа ялгарна.

14-р бүлэгт үндсэнд нь “*н*” гардаггүй бүлгээс өгөх оршихын тийн ялгалын хувиллаар ялгардаг үгс орно.

15-р бүлэгт хос эгшгээр төгссөн үгс багтана. 12-р бүлгээс ялгарах онцлог нь харьяалахын тийн ялгалд “*-н, -г*”, өгөх оршиход “*-д*” нөхцөлөөр хувилдаг.

Кирилл үсгийн дүрэмд хос эгшгээр төгссөн нэр үгийн араас гарахын тийн ялгалын нөхцөлийг залгахад “*г*” гийгүүлэгч жийрэглэн *малгай → малгайгаас, орой → ойройгоос, зай → зайгаас* гэхчлэн бичихээр заажээ. Энэ нь маргаангүй монгол хэлний зүйд нийцсэн бичлэг мөн.

Гэвч, эдүгээ нийтийн хэвлэл мэдээллийн хэрэгсэлд бичиж, аман ярианд хэлэлцэн буйгаас үзэхэд, “*г*” жийргийн хажуугаар, “*н*” жийргийг хэрэглэх нь түгээмэл болжээ. Ийм хэлзүйн үг бичгийн хэлэнд ч нэвтэрсэн байх тул хөрвүүлэг хийх зорилгоос шалтгаалан “*-(н)аас, -(г)аас*“ хоёр хэлбэрийг сонгох боломжтойгоор орууллаа.

16-р бүлэгт хамрын “*н*”-ээр төгссөн үгс орно. Зарим тохиолдолд “*г*” жийрэглэнэ.

17-р бүлэгт эгшигт гийгүүлэгчээр төгссөн, балархай эгшигтэй үгс багтах ба 1-р бүлгээс балархай эгшгээрээ ялгарна.

Жишээ: 17.1 0. *арал, мандал* 1. *арл-, мандл-*

17.2 0. *ноёрхол, бодол* 1. *ноёрхл-, бодл-*
17.3 0. *цөөрөм, нөлөөлөл* 1. *цөөрм-, нөлөөлл-*
17.4 0. *үнэмшил, өгүүлэл* 1. *үнэмшл-, өгүүлл-*

Үндсийн хэлбэр нь балархай эгшигтэй үгсийн тухайд 0 болон 1 гэсэн утгатай. 0 хэлбэр нь балархай эгшиг бүхий хэлбэр, 1 нь балархай эгшгээ гээсэн хэлбэр болно.

18-р бүлэгт “*и*”-ээс бусад богино эгшгээр төгссөн үгс орно. Эгшиг нь ялгах эгшиг байна. Үндсийн 2 хэлбэртэй.

19-р бүлэгт заримдаг гийгүүлэгчээр төгссөн, балархай эгшигтэй үгс хамаарна. 3-р бүлгээс балархай эгшгээрээ ялгарна.

20-р бүлэгт “*н*” гийгүүлэгчээр төгссөн, балархай эгшигтэй үгс багтана. 2-р бүлгээс мөн балархай эгшгээрээ ялгарна.

21-р бүлэгт 5-р бүлгээс балархай эгшгээрээ ялгарах үгс орно.

22-р бүлэгт 6-р бүлгээс балархай эгшгээр ялгардаг үгс хамаатай.

23-р бүлэгт “*эгшигт гийгүүлэгч+зөөлний тэмдэг*”-ээр төгссөн үгс орно.

Жишээ: 23.1 0. *сонгууль, хань* 1. *сонгуули-, хани-*
23.2 0. *лооль, говь* 1. *лооли-, гови-*
23.4 0. *дизель, мебель* 1. *дизели-, мебели-*

Үндсийн хэлбэр нь зөөлний тэмдгээр төгссөн үгсийн тухайд 0 болон 1 гэсэн утгатай. 0 хэлбэр нь зөөлний тэмдэг бүхий хэлбэр, 1 нь зөөлний тэмдэг “и” болсон хэлбэр болно.

24-р бүлэгт “*заримдаг гийгүүлэгч + зөөлний тэмдэг*”-ээр төгссөн үгс орно.

25-р бүлэгт 18-р бүлгээс үндсэнд нь “*н*” гарч ирдгээрээ ялгардаг үгс багтана.

26-р бүлэгт 19-р бүлгээс үндсэнд нь “*н*” гарч ирдгээрээ ялгардаг үгс орно.

27-р бүлэгт 11-р бүлгээс балархай эгшгээрээ ялгардаг үгс хамаарна.

28-р бүлэгт 23-р бүлгээс үндсэнд нь “*н*” гардгаараа ялгаатай үгс багтана.

29-р бүлэгт 24-р бүлгээс үндсэнд нь “*н*” гардгаараа ялгаатай үгс орно.

30-р бүлэгт урт эшгээр төгссөн бөгөөд 12-р бүлгээс харьяалах, гарах, үйлдэх, эзэнд хамаатуулах нөхцөл залгахад “г” жийрэглэж, өгөх орших, чиглэх, биед хамаатуулах нөхцөлийг шууд залгадгаараа ялгардаг үгс багтана.

31-р бүлэгт 4-р бүлгээс балархай эгшгээрээ ялгардаг үгс орно.

32-р бүлэгт “*вч*” дагавраар төгссөн үгс орно.

“*вч*” дагавраар төгссөн үгсийг тусгайлан бүлэглэсний учир нь үсгийн дүрэмд “*–вч*” дагавраар бүтсэн нэр үгийг харьяалах ба өгөх оршихын тийн ялгалаар хэлбэржүүлэхдээ *–ийн, -(и)д* хэлбэрийг залгана хэмээснийг дагахын хамт, бодит хэрэглээнд тогтворгүй “*н*” бүхий үгсийн адилаар хувилах нь түгээмэл болсныг харгалзан, мөн монгол хэлний зүй тогтлоор гийгүүлэгчээр төгссөн үгийн үндсэнд “*н*” гарахгүй, эгшгээр төгссөн үгийн үндсэнд “*н*” гарах боломжтой тул энэ бүлэгт харгалзах залгавар бүтээврийн олонлогийг “*н*” агуулсан болон агуулаагүй гэсэн хоёр бүрэлдэхүүнтэйгээр гаргав.

Үйл үгийн бүлэг

Кирилл үсгийн дүрмээр үйлийн хувилах бүлэг 16. Үгийн бүлэг нь мөн эгшгийн зохицлоос хамааран 4 хүртэл хуваагдана [6].

1-р бүлэгт урт эгшгээр төгссөн үгс орох бөгөөд урт эгшгээр эхэлсэн нөхцөл залгахад “*г*” жийрэглэнэ.

Жишээ:	1.1	*асуу-, бай-*	1.2	*боо-, зохио-*
	1.3	*ерөө-, нөө-*	1.4	*нээ-, хүлээ-*

2-р бүлэгт богино “*и*” эгшиг, туслах эгшгээр төгссөн үгс багтах ба урт эгшгээр эхэлсэн нөхцөл залгахад урт эшгийн нэгийг хасаж бичнэ.

3-р бүлэгт “*м, н, л, в*”-ээс бусад буюу “*р, г*” эгшигт гийгүүлэгчээр төгссөн үгс орно. Дан гийгүүлэгчээс бүтсэн болон давхар гийгүүлэгчээр эхэлсэн нөхцөл залгахад эгшиг жийрэглэнэ.

Өнгөн талдаа 5-р бүлэгтэй ижил мэт боловч тусгайлан бүлэг болсны учир нь гийгүүлэгчээр төгсгөн бичихээр журамласан болохоос монгол бичгийн хэлэнд эгшгээр төгсдөг тул 5-р бүлгээс өөр хувилалтай.

4-р бүлэгт “*м, л, в*” эгшигт гийгүүлэгчээр төгссөн үгс орж, 3-р бүлгээс “*вал*^{8}*, ваас*^{8}*”* нөхцөлийн *“бал*^{4}*, баас*^{4}*”* хэлбэр авдгаараа ялгарна.

5-р бүлэгт эгшигт гийгүүлэгчээр төгссөн үгс багтана. Тус бүлгийн эгшигт гийгүүлэгч нь монгол бичгийн хатуу дэвсгэр гийгүүлэгчтэй дүйнэ. 3-р бүлгээс ялгарах нь нөхцөлдүүлэн холбохын зэрэгцэхийн “*ч*”, цагаар төгсгөхийн “*чээ*” хэлбэрийг авна.

6-р бүлэгт “*эгшиг + заримдаг гийгүүлэгч*”-ээр төгссөн үгс орно.

7-р бүлэгт “*гийгүүлэгч + заримдаг гийгүүлэгч*”-ээр төгссөн үгс хамаарна. 6-р бүлгээс ялгарах нь одоо ба ирээдүй цагаар төгсгөх “*-на*^{4}” нөхцөлийг залгахад зохих эгшгийг жийрэглэнэ.

8-р бүлэгт “эгшиг + “ж, ч, ш” гийгүүлэгч”-ээр төгссөн үгс багтана.

9-р бүлэгт “*гийгүүлэгч + “ж, ч, ш”гийгүүлэгч*”-ээр төгссөн үгс орно. 8-р бүлгээс ялгарах нь одоо ба ирээдүй цагаар төгсгөх “*-на*^{4}” нөхцөлийг залгахад “и” эгшиг жийрэглэнэ.

10-р бүлэгт “*и*”-ээс бусад богино эгшгээр төгссөн үгс багтана. Урт эгшгээр эхэлсэн залгавар нөхцөл залгахад нэг эгшгийг гээж бичнэ. Үүнийг шийдэхдээ “*үндсийн хэлбэр*” ашиглав.

Жишээ:	10.1	0. *дагна-, зарла-*	1. *даг-, зарл-*
	10.2	0. *орло-, оно-*	1. *орл-, он-*
	10.3	0. *зөвлө-, сөнө-*	1. *зөвл-, сөн-*
	10.4	0. *сэлгэ-, хэвлэ-*	1. *сэлг-, хэвл-*

11-р бүлэгт эгшигт гийгүүлэгчээр төгссөн, балархай эгшигтэй үгс орно. Эгшигт гийгүүлэгч нь монгол бичгийн зөөлөн дэвсгэр гийгүүлэгчтэй дүйнэ.

12-р бүлэгт эгшигт гийгүүлэгчээр төгссөн, балархай эгшигтэй үгс хамаарна. Эгшигт гийгүүлэгч нь монгол бичгийн хатуу дэвсгэр гийгүүлэгчтэй дүйнэ.

13-р бүлэгт “*эгшигт гийгүүлэгч + зөөлний тэмдэг*”-ээр төгссөн үгс орох бөгөөд эгшигт гийгүүлэгч болон урт эгшгээр эхэлсэн залгавар нөхцөл залгахад зөөлний тэмдэг “*и*” болж өөрчлөгдөнө.

14-р бүлэгт “*заримдаг гийгүүлэгч + зөөлний тэмдэг*”-ээр төгссөн үгс багтах бөгөөд 13-р бүлгээс заримдаг гийгүүлэгчээр эхэлсэн залгавар нөхцөл залгахад зөөлний тэмдэг нь мөн “*и*” болон хувирдгаараа ялгаатай.

15-р бүлэгт заримдаг гийгүүлэгчээр төгссөн, балархай эгшигтэй үгс багтана.

16-р бүлэгт “*ж, ч, ш*” гийгүүлэгчээр төгссөн, балархай эгшигтэй үгс орно.

Монгол бичгийн дүрмийг бүлгийн кодоор илэрхийлэх нь

Монгол хэл шинжээчид өдий төдий хэлзүйн бичгийг зохион туурвисаар ирсэн бөгөөд монгол зохицсон үг зүйн зарчмыг голлон мөрддөг зөв бичих зүйтэй нягт холбоотой эргэлзээгүй юм.

Монгол бичгийн зөв бичих зүйн тулгуур үндэс нь *“чанга хөндийн зохицол”, “дэвсгэрлэх ёсон”, “нэгэн үеийн дотор эгшиг гийгүүлэгч салаалан орох ёс буюу үе бүтэх ёс”* билээ. Монгол хэлний уугуул онцлог, язгуур хэв шинжийг харгалзан үзсэний үндсэнд хэл шинжлэлийн ул суурьтай, нарийн чанд боловсруулсан тус зөв бичих зүйг программд таниулахад ч харьцангуй хялбар байна.

Тухайлбал, монгол бичгийн зөв бичих зүйн дүрмээр нэр, үйл тус бүр 6 бүлэгт бүрэн багтаж байна [6].

Нэр үгийн бүлэг

1-р бүлэгт эгшгээр төгссөн, тогтворгүй “*н*”-гүй үгс орно.

Жишээ: 1.1 `Hologe%t , DeleI` 1.2 `KiroehkEt , Zoik;O`

`DeleI Nogooa , DeleI iia , DeleI dO , DeleI iI , DeleI fca , DeleI bE* , DeleI DeI , DeleI bEa`

2-р бүлэгт эгшгээр төгссөн, тогтворгүй “*н*” бүхий үгс багтана.

Жишээ: 2.1 `MorI , NebcI` 2.2 `NidO , fidekEt`

`NidO NoKoa , Nidoa O , Nidoa dO , NidO iI , Nidoa fca , NidO bE* , NidO DeI , NidO bEa`

3-р бүлэгт зөөлөн дэвсгэр гийгүүлэгчээр төгссөн үгс хамаарна.

Жишээ: 3.1 `No^ , Meeedege&` 3.2 `frde^ , DoilokEle&`

4-р бүлэгт хэлний үзүүрийн ‘*н*’ (n)-ээр төгссөн үгс орно.

Жишээ: 4.1 `fecidea , Cegea` 4.2 `fjeleKa , CeehkEkEa`

`fecidea Nogooa , fecidea O , fecidea dO , fecidea I , fecidea fca , fecidea iie* , fecidea DeI`

5-р бүлэгт хамрын “*н*” буюу ‘*их инхлэг’* (ng)-ээр төгссөн үгс хамаарна.

Жишээ: 5.1 `Seeit , bEiisieit` 5.2 `fdleeit , KiBeit`

6-р бүлэгт хатуу дэвсгэр гийгүүлэгчээр төгссөн үгс багтана.

Жишээ: 6.1 Geje* , feiime(6.2 bEleit , foikE*

Үйл үгийн бүлэг

1-р бүлэгт эгшгээр төгссөн үгс орно.

Жишээ: 1.1 fesegO, Hera 1.2 kuciya, kEla fesegojO , fesegogeoa , fesegoa , fesegoeesege* , fesegobEcO , fesegomeeeca , fesegode&T , fesegoeeole* , fesegobE& , fesegobEsO , fesegomejia , fesegomejI , fesegoehgoda , fesegogoda , fesegoeh)T , fesego& foikEI, fesegoeeo&T , fesegoroa , fesego*T

2-р бүлэгт зөөлөн дэвсгэр гийгүүлэгчээр төгссөн үгс хамаарна.

3-р бүлэгт хатуу дэвсгэр гийгүүлэгчээр төгссөн үгс багтана.

4-р бүлэгт ‘*б*’ (b) гийгүүлэгчээр төгссөн 3 үг орно.

5-р бүлэгт ‘*н*’ (n) гийгүүлэгчээр төгссөн 4 үг хамаарна.

6-р бүлэгт ‘*г*’ (g, γ) гийгүүлэгчээр төгссөн 2 үг орно.

Эцэст нь тэмдэглэхэд, хөрвүүлэх программд монгол, кирилл бичгийн зөв бичих зүйг таниулах явцад олон асуудал илэрч гардаг бөгөөд энэхүү алгоритмчилах ажил нь орчин цагийн монгол хэлний зүй тогтол, зарим гажуу хэвшсэн хэлбэр, засууштай гажсан хэлбэр зэргийг өргөн хүрээнд бүртгэн судлах боломж олгож байна. Орчин цагийн монгол хэлний хэлзүйд ерөнхий зүй тогтлыг цөөн хэдэн жишээгээр авч үзсэн байдаг бол идэвхтэй хэрэглээнд буй үг бүр дээр шалган үзэхэд, өдий төдий бэрхшээл гарч ирдэг.

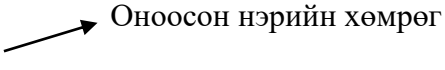
1.3 Дэд сангууд

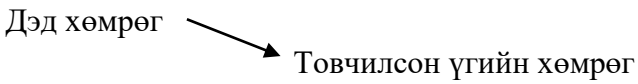
Өмнө дурдсанчлан нэр, үйл үгийн хувилал бүхий хэлбэрийг оруулаагүй 50 мянга гаруй үгийн үндсийг агуулж буй үндсэн сан дээр нэмж, оноосон нэр, товчилсон үг болон зарим нэгэн онцой үгсийн сан (нэрлэсэн нэгж болоод бичлэгийн өвөрмөц онцлог бүхий үгс) бүхий тусгай санг бий болгох шаардлага урган гарсан юм.

Эдгээр дэд сангууд нь тэдгээр үгсийг кирилл болон монгол бичгийн дүрмийн дагуу машинаар зөв бүтээж мөн задлах зорилгыг хангана.

Дэд санг үүсгэх зайлшүй нэг чухал зүйл нь оноосон нэрийг хувилгах кирилл үсгийн дүрэмтэй холбоотой. Жишээ нь: “баатар” гэдэг үг ерийн нэр бол “баатрын” оноосон нэр бол “Баатарын” гэж хувилах ёстой.

Одоогийн байдлаар 1500 гаруй үг бүхий товчилсон үгийн сан ба 260 мянга гаруй үг бүхий оноосон нэрийн сан бүрдүүлсэн байна. Тиймээс хөрвүүлэх программд зориулсан одоогийн хувилбарын дэд сангийн бүтцийг дараах байдлаар илэрхийлж болно.





Зураг 4. Хөмрөгийн ангийлал

Эдгээр дэд хөмрөгийг байгуулах болсон бодот шалтгаан, дэд хөмрөгийн нэгжид тавих шалгуур, түүний шийдлийн талаар товч өгүүлье.

Оноосон нэрийн сан

Оноосон нэр бол оногдохуунаасаа шалтгаалан тухайн ард түмний зан заншил, уламжлал, ертөнцийг үзэх үзэл, байгаль, нийгмийн орчин, хэл сэтгэхүйнх шүтэн барилдлагыг тусгасан, тодорхой бүтэц, утга бүхий нийлэмж үгээс бүтсэн цогц ухагдахуун билээ. Тиймээс профессор Равдан “*Оноосон нэр нь нийгэм дэх тодорхой хүмүүсийн бүтээл нэгэнт мөн...*” [Равдан, 2008а, х.7] хэмээжээ.

Ерийн нэр нь оноосон нэр, түүний гишүүн болохдоо тодорхой оногдохууныг заасан шинэ харьцаанд орно. Тэрхүү оноосон нэр тухайн юмаа орон зай, цаг хугацааны оршихуйнх нь хүрээнд заах гол үүрэгтэй. Тиймээс оноосон нэрийг оногдохуунаас нь ангид авч үзэх боломжгүйн дээр тогтолцоот чанартай нь уялдан шинжлэх ухаанууд судалгааныхаа зорилгын үүднээс судлагдахуунаа болгож иржээ. Тухайлбал, геологич, газар зүйчид газрын хэвлий дэх эрдэс баялгийн судалгаанд нэрийг нь баримжаа болгож, түүхчид өнгөрсний улбааг тэрхүү нэрээр дамжуулан мөшгиж, угсаатны зүйч, хүн судлаачид угсаатны хэв байдлыг судлан шинжлэхдээ чухалчилдаг.

Хөрвүүлэх программын хувьд оноосон нэрийг тусгайлан дэд хөмрөг болгохдоо тэдгээр нэрийг кирилл болон монгол бичгийн дүрмийн дагуу машинаар зөв хөрвүүлэх зорилгоос үүдэлтэй юм. Энэхүү цахимд зориулан оноосон нэрийг боловсруулах арга, хандлагыг тайлбарлая.

Компьютер нь хоосон зайгаар тусгаарлагдсан л бол “*нэг юм*” гэж үздэг учраас хамт бичигдэж байгаа бүхнийг нэг үг буюу “*нэгж*” гэж тооцох хэрэгтэй болно. Тэгэхээр оноосон нэрийн гишүүн болсон нэр нь тусдаа бичигдэж буй л бол харьяа хөмрөгийнхөө нэгжид тооцогдоно гэсэн зарчмыг баримтлах болж байгаа юм.

Оноосон нэрийн, ялангуяа хүний нэрийн дийлэнх олонх нь хоёр нэрээс, заримдаа түүнээс олон нэрээс бүтсэн байдаг. Энэхүү нийлмэл бүтэцтэй байдаг онцлог шинж нь дэд хөмрөг болгох гол шалтгааны нэг юм.

Дараагийн чухал зүйл бол кирилл үсгийн “Оноосон нэрийн төгсгөлийн гийгүүлэгчийн өмнөх балархай эгшгийг гээхгүй” гэсэн дүрэмтэй холбоотой. Энэ дүрмийн дагуу зарим үг хам сэдэвт ерийн нэр эсвэл оноосон нэр болж байгаагаасаа үүдэн хэдий нэг үг боловч цаашид өөр өөр хэлбэрээр хувирах болно. Ингэхлээр тухай бүрт нь хувиллын бүлгийн өөр өөр код харгалзуулах ёстой гэсэн үг.

Уг дүрмээс шалтгаалан нэг нэрээс бүтсэн оноосон нэрийг тус дэд хөмрөгт багтаах шаардлагатай болов.

Дээрх учир шалтгааны мөрөөр оноосон нэрийн хөмрөгийн нэгжид дараах шалгууруар хандлаа. Үүнд:

1. Хөмрөгт “*нэг нэгж байх*”

2. Нэгж “*зөв байх*”
    а. хүчин төгөлдөр зөв бичих дүрмийг мөрдөх
    б. тухайн нэрийн утгыг харгалзах зэрэг болно.

1.4 Бүрдүүлсэн сангийн тухай

Хүснэгт 1. Сангийн тоо

№	Сангийн нэр		Тоо хэмжээ	
			Кирилл бичиг	Монгол бичиг
1	Нөхцөлийн сан	Нэр үг	Давхардсан тоогоор – 35,932	Давхардсан тоогоор – 8,983
		Үйл үг	Давхардсан тоогоор – 1,425,967	Давхардсан тоогоор – 365,490
2	Үг	Үгийн үндэсийн сан	52,784	52,784
		Мэдээний сан	-	Давхардсан - 179,064 Давхардаагүй - 20,179
		2 сая	-	Давхардсан - 60,578,634 Давхардаагүй - 2,266,882
		Өдрийн сонин	Давхардаагүй тоогоор - 38,695	-
		108 боть	Давхардаагүй тоогоор - 166,955	-
		Холбоо үгийн сан	Давхардсан тоогоор - 30,050	Давхардсан тоогоор - 30,050
4	Өгүүлбэр	2 сая	-	2,569,794
		Мэдээний сан	-	8,349
		Өдрийн сонины сан	25,946	-
		108 боть	268,898	-
		Холбоо үгийн сан	11,868	11,868
5	Холбоо үгийн сан		77,597	77,597
6	Оноосон нэрийн сан		227,418	227,418
7	Товчилсон үгийн сан		1100	-

Хүснэгт 2. Нөхцөлийн сан

Нэр үг (нөхцөл давхарлан орсон тоо)	Үйл үг үг (нөхцөл давхарлан орсон тоо)
▪ 2 давхар – 1,244ш ▪ 3 давхар – 5,808ш ▪ 4 давхар – 11,600ш ▪ 5 давхар – 17,280ш	▪ 2 давхар – 4,224 ▪ 3 давхар – 30,716 ▪ 4 давхар – 139,465 ▪ 5 давхар – 396,684 ▪ 6 давхар – 282,176 ▪ 7 давхар – 265,782 ▪ 8 давхар – 161,480 ▪ 9 давхар – 145,440
Нийт 35932 ширхэг боломжит хосолсон үр дүн гарсан.	Нийт 1425967 ширхэг боломжит хосолсон үр дүн гарсан.

Зураг 5. Нөхцөлийн сангийн жишээ

Хүснэгт 3. News.mn сайтаас бүрдүүлсэн мэдээний сан

Төрөл	Нийт мэдээний тоо	Нийт өгүүлбэрийн тоо	Нийт үгийн тоо	Давхцаагүй үгийн тоо	Хугацаа
Спорт	287	1,580	28,173	2,838	2018.08.16 – 2019.05.14
Нийгэм	285	1,392	31,288	3,209	
Соёл, Урлаг	285	1,447	33,416	4,314	
Улс төр	288	2,329	47,879	5,260	
Эдийн засаг	279	1,601	38,308	4,558	
	1,424	8,349	179,064	20,179	

News.mn сайт нь 2018.08.16 өдрөөс эхлэн спорт, нийгэм, соёл, улс төр, эдийн засаг гэсэн 5 төрлийн мэдээг монгол бичгээр давхар хөтлөх болсон. Энэ өдрөөс хойш 2019.05.14 өдрийг хүртэлх бүх мэдээг авч үг болон өгүүлбэрийн санг бүрдүүлсэн. Энэ хугацаанд нийт 8,349 өгүүлбэртэй, давхардсан тоогоор 179,064 үгтэй, давхардаагүй тоогоор 20,179 үгийн тусламжтайгаар 1,424 мэдээг нийтэлсэн байна.

Хүснэгт 4. News.mn сайтаас бүрдүүлсэн n-gram-ийн тоо

Өгүүлбэрийн сан	trigram	bigram	monogram
8,349	80,410	103,672	20,179

Хүснэгт 5. Итгэлт хамбын тухай материалаас бүрдүүлсэн сан

Итгэлт хамба	Хуудасны тоо	Өгүүлбэрийн тоо	Үгийн тоо
1	92	1577	38582
2	113	1896	47797

Өдрийн сонины дугаар	Хуудасны тоо	Өгүүлбэрийн тоо	Үгийн тоо
2018 оны 01-р сар	1125	26218	417704
2018 оны 02-р сар	752	19420	329224
2018 оны 03-р сар	841	21799	357479

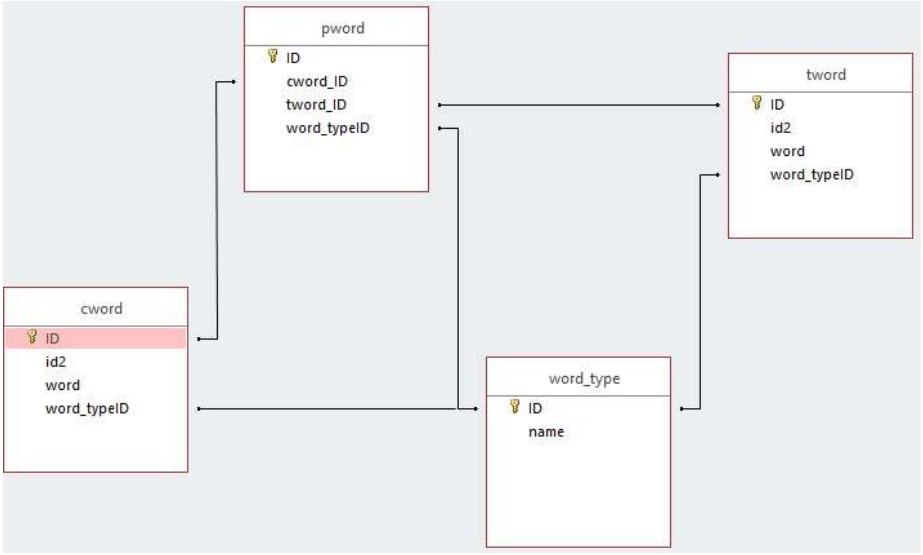
Хүснэгт 6. Өдрийн сониноос бүрдүүлсэн мэдээ, нийтлэлийн сан

Өгүүлбэрийн сан	trigram	bigram	monogram
268899	2723633	1807234	166955

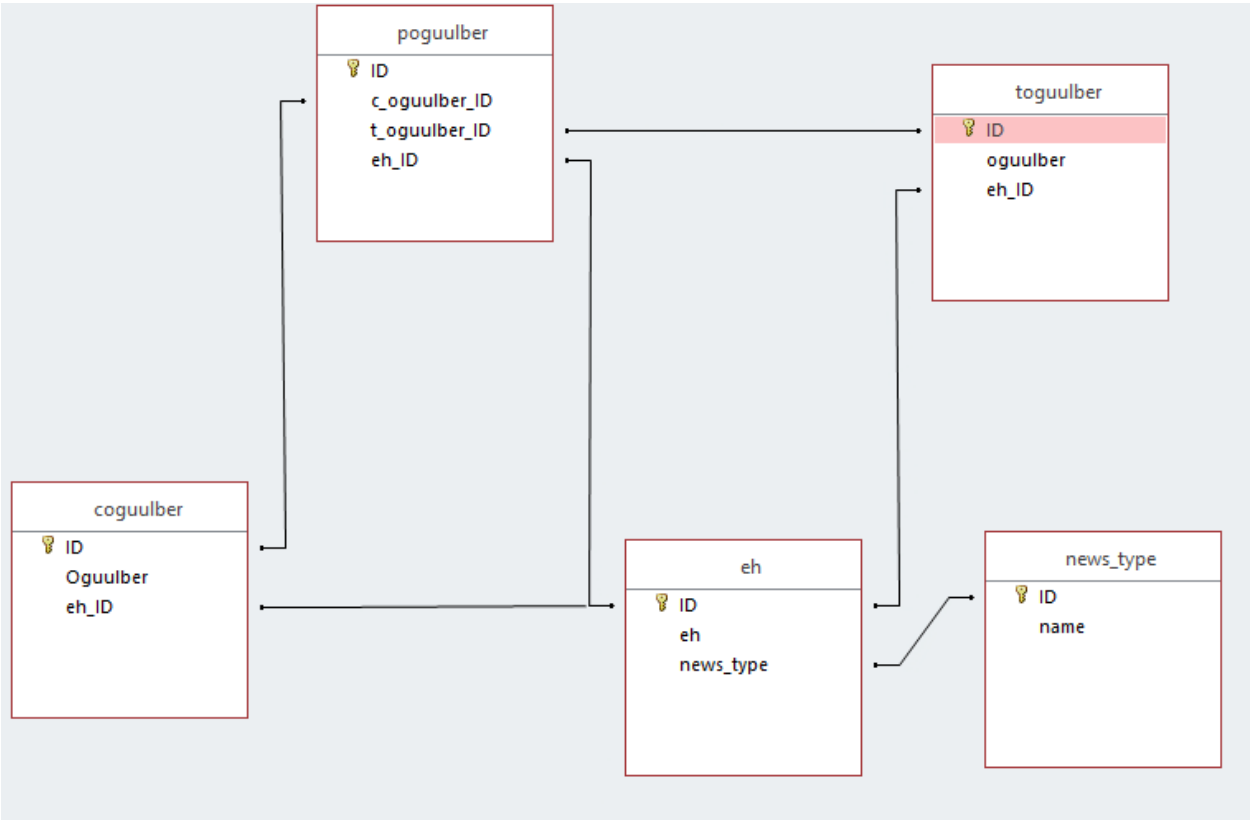
Хүснэгт 7. Монголын уран зохиолын дээжис 108 ботиос бүрдүүлсэн сан

ӨМИСургуулиас монгол бичгийн 2 сая өгүүлбэрийг авч үгийн сан болгосон. Эдгээр өгүүлбэрээс давхцсан байдлаар 60,578,634 үг, давтагдаагүй байдлаар 2,266,882 үг байсан бөгөөд үүнээс хамгийн олон давтагдсан үг нь “᠂” - 892550, “᠋᠋” – 500111 удаа давтагдсан байна.

Хоёр бичгийн хооронд хөрвүүлэх программд ажиллаж буй сангийн бүтэц, зохион байгуулалт.



Зураг 6. Үгийн сангийн бүтэц



Зураг 7. Өгүүлбэрийн сангийн бүтэц

1.5 Бүлгийн дүгнэлт

Компьютер хэл шинжлэлийн судалгаанд зориулан тусгай зориулалтын төрөл бүрийн өгөгдлийн сангуудыг зохион байгуулдаг. Ажлын явцаас шалтгаалан төрөл зүйл, дотоод бүтэц, зохион байгуулалт, бүрдүүлэх аргатай холбоотой асуудалуудыг тусгайлан судалдаг Corpus Linguistics гэсэн салбар ч бий болсон.

Энэ төслийг гүйцэтгэхийн тулд монгол хэлний үг зүйг судалж, монгол хэлний хувьд дүрмийн болон үгийн сангийн файлыг зохих шаардлагад нийцүүлэн үүсгэх зайлшгүй шаардлага тулгарсан. Тухайлбал монгол үгсийн үгийн аймаг, хувилалын хэлбэр зэргийг тогтоон, улмаар шаардлага хангасан өгөгдлийн сангуудыг үүсгэсэн. Өмнө дурдсанчлан нэр, үйл үгийн хувилал бүхий хэлбэрийг оруулаагүй 79000 гаруй үгийн үндсийг агуулж буй үндсэн сан дээр нэмж, оноосон нэр, товчилсон үг болон зарим нэгэн онцой үгсийн сан, тусгай зориулалтын сангуудийг бий болгох шаардлага урган гарсан юм. Эдгээр сангууд нь үгсийг кирилл болон монгол бичгийн дүрмийн дагуу машинаар зөв бүтээж мөн задлах, бичвэрээс алдааг илрүүлж засах, бичвэр хооронд хөрвүүлэх туршилт хийх зорилгыг хангана.

Монгол хэлний нь олон тоо, тийн ялгал, хамаатуулах, үгүйсгэх, хэв, байдал, нөхцөлдүүлэн холбох, биеэр төгсөх, тодотгон холбох, цагаар төгсөх зэрэг нөхцлүүдийг багтаасан кирилл, монгол бичгийн хэлбэрүүдийг агуулсан нөхцөлийн сан үүсгэсэн. Монгол хэлний нэг үгийн бүтцэд хэд хэдэн нөхцөл давхарлан орж болно. Ер нь монгол хэлний үгийн бүтцийг ажиглаж үзэхэд 3-4 давхарлан орсон тохиолдол байна. Нөхцөл давхарлан орох дэс дараа нь нарийн зүй тогтолтой. Үгийн бүтцэд оролцож байгаа бүтээврүүд нь тогтсон байр, нарийн дэс дараатай бөгөөд тэдгээрийн зааг нь тодорхой байдаг. Иймд нөхцөл давхарлан орох өгөгдлийн санг кирилл болон монгол бичгийн хувьд бүрдүүлсэн.

Мөн эдгээр өгөгдлийн сангуудыг бүрдүүлэх ажлын явц, сангийн ач холбогдол, хэрэглээ, өөрсдийн гүйцэтгэсэн туршилт зэргээс үндэслэн дараах дүгнэлтүүдийг хийж байна.

1. Судалгааныхаа ажилд зориулан кирилл болон монгол бичгийн тус бүрийнх нь онцлогийг хадгалж чадсан тусгай зориулалтын өгөдлийн сангуудыг зохион байгуулсан. Энэхүү өгөгдлийн санд үгийн монгол, кирилл бичгээрх зохистой бичлэгийг оруулсан бөгөөд хоёр бичгийн зөв бичих дүрмийн тогтолцоог компьютерд таниулах аргачлалыг боловсруулан багтаасан зэрэг нь шинэлэг тал болно. Компьютер хэл шинжлэл, монгол хэлний кирилл болон монгол бичгийн бичвэрийг цахимаар боловсруулах (үг зүйн хувилал, өгүүлбэр зүйн задлагч, кирилл ба уламжлалт монгол бичгийн хөрвүүлэгч болон зөв бичих) чиглэлээр цаашид хийж гүйцэтгэх олон ажлын суурь нь бидний бүрдүүлсэн энэ сан байх болно.
2. Монгол үгийн бүтэц, үг бүтэх ёс, үгийн аймаг, бүтээвэр болон үгийн хувилал, кирилл болон уламжлалт монгол бичиг тэдгээрийн онцлог, ялгаатай болон адилхан шинж чанарын талаар судлаж монгол хэлний онцлогоос хамааруулан өгөгдлийн сангаа зөв зохистой зохион байгуулах хэрэгтэй. Эдгээрийн үндсэн дээр дараах дүгнэлтүүдийг хийлээ.
   - Үгийн язгуурт дагавар залгаж монгол хэлний зөв үг бүтээх эсэх нь монгол хүний ухамсарт түүний тусах байдлаас хамаардаг тул өгөгдсөн язгуураас бүтээгдэх бүх

үндсийг алгоритмаар үүсгэх боломжгүй. Иймээс үгийн санд идэвхитэй язгууруудыг оруулахаас гадна идэвхигүй ба идэвхитэй язгуураас дагаврын аргаар үүсэх үгийн үндсүүдийг үгийн санд оруулна.

- Үгийн аймаг нь монгол хэлний ойлголт тул кирилл, монгол бичигт ижил байна. Монгол хэлний үгийн үндсийн хувилах нөхцөлийг тодохойлох зорилгоор нэр үг, үйл үг, үл хувилах үг гэж ангилж болох юм. Нэр ба үйл үгийн нөхцөл нь тухайн аймгийн бүх үгэнд залгаж болдог бүтээлч шинж чанартай тул нөхцөлийг дугаарлаж, кирилл ба монгол бичгийн хэлбэрүүдийн хамт нэг санд хадгална. Ингэснээр нөхцөлийг хөрвүүлэхэд хялбар болно.
- Үгийн бүтцэд оролцож байгаа нөхцөлүүд нарийн тогтсон дараалалтай байдаг. Монгол хэлэнд нөхцөл давхарлан орох хэлний үзэгдэл нь кирилл болон монгол бичигт хоёуланд нь биелнэ. Энэхүү нөхцөлүүдийн дараалан орох боломжийг тусад нь сан үүсгэн хадгална. Монгол хэлэнд үг нөхцөлөөр хэрхэн хувилах болон үгэнд нөхцөлийн ямар хэлбэр залгах нь үгийн төгсгөлөөс хамаарна. Өөрөөр хэлбэл монгол хэлэнд үг нөхцөлөөр хувилах нь үгийн төгсгөлөөс хамаарна. Кирилл, монгол бичигт ижил төгсгөлтэй үгүүдэд ижил хэлбэрийн нөхцөл залгагдаж, ижил хэлбэрээр хувирдаг байна. Үгэнд нөхцөл залгах дүрмийг кирилл болон монгол бичигт тус тусад нь авч үзнэ.
- Олон хувилбар бүхий үндэстэй үгсийн хувиллуудыг тус тусад нь үгийн санд оруулана. Өөрөөр хэлбэл кирилл бичлэг нь ижил боловч монгол бичигт утгаас хамаарч ялгаатай бичдэг үгсийг үгийн санд монгол бичгийн ялгаатай бичдэг хэлбэр бүрээр оруулж, утгын тайлбарыг бичнэ. Монгол бичгийн хэлбэр бүрийг дугаарлаж, кириллээс монгол бичигт хөрвүүлэх үед аль хэлбэрээр байхыг утга таних аргаар тодорхойлно.
- Цаашид нэгэнт буй болгосон тэмдэглэгээ бүхий өгөгдлийн санг ашиглан үгийг хувилалтай нь үүсгэж, түүнийгээ интернет болон бодит хэрэглээн дээрх эх бичвэрүүдтэй тулгах замаар зарим зарим асуудлуудыг цахим аргаар шийдэх боломжтой.

3. Энэхүү сангийн ач холбогдлын талаар олон зүйл хэлж болох боловч ерөнхийд нь тун товчхоноор дараах байдлаар илэрхийлж болох юм. Компьютер хэл шинжлэлийн талаар цаашид бидний хийх олон ажлын суурь нь энэ сан байх болно. Тухайлбал:
   - Монгол хэлийг компьютерээр боловсруулахад зориулсан үг зүйн загварчлал, өгөгдлийн сангийн зохиомжийг боловсруулж, түүнд тохирсон өгөгдлийг бүрдүүлсэн.
   - Эдгээр өгөгдлийн сангийн бүтэц бүрэлдэхүүн нь одоогоор хамгийн өргөн хүрээг хамарч буй.
   - Монгол хэлний үг зүйн боловсруулалт хийнэ.
   - Кирилл монгол бичгийн хооронд цахим хөрвүүлэг хийнэ.
   - Зөв бичлэг шалгуур (spell checker)-ыг монгол болон кирилл хоёр бичгийн хувьд зохиох бүрэн үндэс болно. Одоогийн ажлын явцад хоёр бичгийн хувьд хамгийн түгээмэл гаргадаг алдааг судлах, үгийн сангийн болон дүрмийн түвшинд алдаа таниулах боломж зэргийг судлах.

- Ирээдүйд монгол хэлний цахим орчуулга хийх программ бүтээх үед үгийн сан болон зөв бичих дүрмийн мэдлэгийн хувьд чухал хэрэглэгдэхүүн болно.
- Бидний байгуулсан өгөгдлийн сан нь цаашид монгол хэлийг компьютерээр боловсруулах бусад ажлыг хийж гүйцэтгэх үндэс суурь нь болно.

II. МОНГОЛ ХЭЛНИЙ ҮГ ЗҮЙН ЗАГВАРЧЛАЛ

Хүн төрөлхтний хэл нь тодорхой тооны нэгж хэсгүүдээс бүрдэх бөгөөд тэдгээр нь нарийн зүй тогтлоор холбоотой бүхэл бүтэн дохионы тогтолцоо юм. Ийм нарийн тогтолцоотой хэлний нэгжүүд нь нэг нь нөгөөтэйгөө эсрэгцэхийн зэрэгцээ бас бие биеэ нөхөж, давхарлан шаталж тогтсон байдаг. Өөрөөр хэлбэл, дээд түвшний нэгж нь доод түвшний нэгжээ багтаасан, доод түвшний нэгж нь дээд түвшний нэгждээ багтсан шинжтэй байна. Жишээлбэл, авиалбар, бүтээвэр, үг, холбоо үг, өгүүлбэр, эх нь хэлний үндсэн нэгжүүд бөгөөд авиалбар нь бүтээврийн бүтцэд оролцож хүртэгдэх, утга ялгах үүрэгтэй хэлний хамгийн бага материаллаг нэгж болно. Бүтээвэр нь үг бүтээх ба хувилгах үүрэгтэй хэлний хамгийн бага утгат нэгж, үг нь холбоо үгийг бүтээх үүрэгтэй хэлний нэрлэлтийн нэгж, холбоо үг нь өгүүлбэр бүтээх үүрэгтэй хэлний нэрлэлтийн дэлгэрэнгүй нэгж, өгүүлбэр нь тодорхой үйл явдлыг илэрхийлэх, нэрлэх, мэдээлэх, харилцах үүрэгтэй хэлний харилцааны бага нэгж, эх нь тодорхой хэрэг явдлын тухай мэдээлэх, харилцах үүрэгтэй харилцааны дээд түвшний нэгж болно. Хэлний нэгжүүдийн гол цөм нь үг бөгөөд хэлний аль ч салбарт үгийг олон талаас нь өөр өөр зорилгоор судалдаг. Үүнээс шалтгаалж хэл шинжлэлд авиазүй, авиалбарзүй, бүтээвэрзүй, үг бүтэх ёс, хэлзүй, үгсийн сангийн судлал, нэрзүй зэрэг салбар ухаанууд үүсчээ.

Үг зүй нь үгсийн дотоод бүтцийг судалдаг ба үг нь бүтээврүүдээс (morpheme) тогтоно. Бүтээвэр гэдэг нь үгийн бүтцийн цааш үл задрах хамгийн бага утгат нэгж юм [5]. Бүтээвэр нь хэлбэр, агуулгын нэгдэлтэй байна. Хэлбэр нь бүтээврийн гадна тал буюу бичих болон хэвлэх үсгийн дүрслэлтэй холбоотой бол утга нь бүтээврийн дотор тал буюу утгатай холбоотой. Дангаараа утга илэрхийлж чадах язгуур бүтээврүүдийг чөлөөт бүтээвэр гэнэ (Free morpheme). Язгуур бүтээвэртэй хамтарч утга илэрхийлэх бүтээврүүдийг нөхцөлт бүтээвэр гэнэ (Bound morpheme).

2.1 Үг зүйн загварчлал

Үг зүй нь үгсийн дотоод бүтцийг судалдаг ба үг нь бүтээврүүдээс (morpheme) тогтоно. Бүтээвэр гэдэг нь үгийн бүтцийн цааш үл задрах хамгийн бага утгат нэгж юм [5]. Бүтээвэр нь хэлбэр, агуулгын нэгдэлтэй байна. Хэлбэр нь бүтээврийн гадна тал буюу бичих болон хэвлэх үсгийн дүрслэлтэй холбоотой бол утга нь бүтээврийн дотор тал буюу утгатай холбоотой.

Дангаараа утга илэрхийлж чадах язгуур бүтээврүүдийг чөлөөт бүтээвэр гэнэ (Free morpheme). Язгуур бүтээвэртэй хамтарч утга илэрхийлэх бүтээврүүдийг нөхцөлт бүтээвэр гэнэ (Bound morpheme).

Бүтээврийн ангилал

Дэлхийн олон хэлний үгийн бүтцэд оролцож байгаа бүтээврийг ангилах ерөнхий зарчим байдаг. Энэ нь ямар ч хэлний бүтээврийн утга, үүрэг, байрын түгээмэл шинж дээр тулгуурласан зарчим юм. Ерөнхий хэл шинжлэлийн онолын үүднээс юуны өмнө утга, үүргээр нь язгуур, залгавар гэж хоёр ангилна. Язгуур нь үгийн сангийн хамгийн ерөнхий (цөм) утгыг хадгалсан цаашид үл задрах үндсэн бүтээвэр юм. Харин үгийн үндэс гэдэг нь

залгавар бүтээвэр авч хувилж болдог бүтээвэр юм. Залгавар бүтээвэр нь язгуур болон үндэс бүтээвэрт залгагдаж шинэ үг бүтээдэг.

Залгавар нь язгуурын утгыг үгийн сан, хэл зүйн төрөл бүрийн утгаар нэмэн дэлгэрүүлдэг туслах бүтээвэр юм. Язгуургүйгээр залгавар үүргээ гүйцэтгэж чадахгүй. Залгавар бүтээврийг язгууртай хэрхэн байрлах дээр нь тулгуурлаж угтвар, оруулбар, дагавар хүрээлбэр гэж ангилдаг. Угтвар нь үгийн язгуур эсвэл үндсийн өмнө, дагавар нь хойно, оруулбар нь дотор, хүрээлбэр нь өмнө ба хойно нь залгагдана.

Угтвар (префикс) гэдэг нь язгуурын өмнө залгаж нэмэлт утга илүүлэх залгавар бүтээвэр юм. Энэ Энэтхэг-Европын бүлэг хэлнүүдэд элбэг тохиолдоно.

Оруулбар (инфикс) гэж язгуурын бүтэц дотор оруулдаг бүтээврийг хэлнэ. Эртний латин, тагаль зэрэг хэлэнд тохиолддог.

Дагавар (постфикс) гэдэг нь язгуурын дараа орж нэмэлт утга илтгэдэг залгавар бүтээврийг хэлнэ. Энэ нь монгол, түрэг, манж-түнгүс зэрэг алтайн хэлний үндсэн залгавар юм. Залгавар бүтээврийг үүргээр нь үг бүтээх, үг хувилгах гэж хоёр ангилдаг. Монгол хэл нь үгийн бүтэц хэв шинжээрээ залгамал хэл учраас бүтээвүүд нь тогтсон тодорхой байр, нарийн дэс дараатай, тэр нь утга, үүрэгтэйгээ шүтэлцсэн байдаг.

Нэг дагавар бүтээвэр олон хэлбэртэй байхыг алломорф /allomorphs/ үзэгдэл гэнэ. Жишээ нь: тодотгон холбохын өнгөрсөн цагийн нөхцөл "сан", "сэн", "сон", "сөн" хэлбэрүүдтэй. Харин ижил хэлбэртэй боловч ялгаатай утга агуулгатай бүтээврийн үзэгдлийг синкретизм гэнэ. Жишээ нь: "аа" нөхцөлийн хэлбэр хамаатуулах болон бусдаар үйлдүүлэх хэвийн нөхцөлд байдаг.

Хэл шинжлэлд хэлийг дараах байдлаар ангилдаг байна [6] [7]. Үүнд:

- Салангид хэл /isolating language/: Жишээ нь: Хятад хэл, Вьетнам. Нөхцөлийн хэлбэрүүд байхгүй бөгөөд үг зүйн үндсэн үйлдэл нь зөвхөн холбоо үг үүсгэх юм.
- Залгамал хэл /agglutinative language/: Жишээ нь: Монгол, Турк хэл. Нөхцөлийн хэлбэрүүд нь угтвар эсвэл дагавар байдаг.
- Хувирдаг хэл /inflectional language/: Жишээ нь: Орос хэл, энэтхэг-европын хэлнүүд. Нэг нөхцөлийн хэлбэр нь ялгаатай хэлбэрүүдтэй байдаг: Нэг бүтээврээр нэгэн зэрэг хэлний олон мэдээллийг өгдөг.
- Нийлмэл хэл /polysynthetic language/: Жишээ нь: Юпик хэл /Yupic/ Төв аляск. Өгүүлбэрийн гишүүд болон нөхцөл нь холилдон залгагдсан байдаг.

Бүх хэлэнд үг үүсгэх процессыг дараах байдлаар ангилдаг байна [8]. Үүнд:

Залгаврын арга /affixes/

- Угтвар /prefix/ - үгийн үндсийн өмнө залгана.
- Дагавар /suffix/ - үгийн үндсийн хойно залгана.
- Оруулбар /infix/ - үгийн үндсэн дотор ордог.
- Хүрээлбэр /circumfix/ - үгийн үндсийн өмнө болон хойно залгана.

Нөхцөл товчлох арга. /cliticisation/ Бие даасан авиа болохгүй, үгийн өмнө болон хойно нь залгагдаж нөхцөлөөр хувилгах, нөхцөлийн хураангуй болсон хэлбэр.

- Клитик бүр бүрэн ба хураангуй хэлбэртэй байна.
- Угтвар клитик - үгийн өмнө залгах клитик
- Дагавар клитик - үгийн хойн залгах клитик

Нийлмэл үг ба холбоо үгийн арга. /compounding and composition/ - хоёр ба түүнээс дээш тооны үг нийлж шинэ үг үүсгэх.

- Хоорондоо зайгүйгээр нийлэх
- Хоосон зай эсвэл дундуур зураас жийрэглэж нийлэх

Авиа давхарлах арга. /reduplication/ - үгийн хэсгийг эсвэл тухайн үгийг бүтэн давтан шинэ үг үүсгэх

- Бүтэн давтах
- Хэсэглэн давтах

Үгийн эгшиг өөрчлөх арга /internal change/

Нэг үгийн хоорондоо ижил төстэй чанаргүй хувиргалтын арга /suppletion/

Хэсэглэх /clipping/, хураангуйлах /abbreviation/, товчилсон нэр/acronymy/

Үг холих арга /blending/

Шинэ үг бүтээх арга /backformation/

Эдгээр бүх үг үүсэх процессуудыг дараах үндсэн 2 салбарт хуваадаг [6] [7] [8].

1. Хувирдаг үг зүй /Inflectional morphology/

Үндсээс үг хувилгах нөхцөлөөр хувилсан үг үүснэ. Жишээ нь:

cat + олон тоо ⇒ *cats*

ном + гарахын тийн ялгал ⇒ номоос

Үүнд: залгаварын, нөхцөл товчлох, үгийн эгшиг өөрчлөх, нэг үгийн хоорондоо ижил төстэй чанаргүй хувиргалтын аргууд хамаарна.

2. Үүсмэл үг зүй /Derivational morphlogy/

Үгийн үндэс эсвэл язгуурт шинэ үг үүсгэх дагавар залган шинэ үг бий болно.

Үүнд: нийлмэл үг ба холбоо үгийн, авиа давхарлах, хэсэглэх хураангуйлах товчилсон нэрийн, үг холих, шинэ үг бүтээх аргууд хамаарна.

Мөн хэлнүүдэд дараах үг зүйн үзэгдлүүд байдаг [9]. Үүнд:

1. Эгшиг зохицох ёс /Vowel harmony/

Үгийн үндэс эсвэл үгийг хувилгасан нөхцөлийн хэлбэр дэх эгшиг хувилгаж буй нөхцөлийн эгшгийг тодорхойлох үзэгдэл.

2. Эгшиг шилжилт /Internal vowel harmony/

Тухайн үгэнд залгах дагавар нөхцөлөөс хамаарч үгийн үндсэнд байсан эгшиг өөрчлөгдөх үзэгдэл. Монгол хэлэнд 1951, 1964 оны монгол хэлний сурах бичиг, хэл шинжлэлийн нэр томъёоны бичигт гээгдэх эгшгийг шилжих эгшиг гэдэг байжээ [2]. Иймээс энэ үзэгдэл нь эгшиг гээгдэх, жийрэглэхтэй төстэй ойлголт юм.

Тэг бүтээвэр /Zero morphlogy/

Бүтээврийн тэг шинж чанарын тухай ойлголт нь Энэтхэг-Европын хэлэнд төдийгүй, бусад бүх хэлэнд байдаг [5]. Хэлний аль ч түвшний нэгж нь ил, далд (тэг) хоёр хэлбэртэй. Ил хэлбэрийн үндсэн дээр далд хэлбэрийг таньж мэднэ. Монгол хэлэнд нэрлэхийн тийн ялгалын нөхцөл тэг хэлбэртэй байдаг. Өөрөөр хэлбэл бүтээврийн хэлбэр нь тэг, агуулга нь нэрлэхийн тийн ялгал гэсэн үг. Жишээлбэл: ах-0 ирлээ.

Үгийн бүтэц: Морфотактик /morphotactic/

Үгэнд бүтээврүүд хэрхэн дараалан орсон болохыг заадаг бүтцийг үгийн бүтэц буюу морфотактик гэнэ. Жишээлбэл:

язгуур + дагавар + олон тооны нөхцөл + тийн ялгал + хамаатуулах

магт + аал + ууд + аар + аа

Хэлэнд бүтээврүүдийн тогтсон ерөнхий байр, дараалал байдаг [5]. Эдгээр байр, дарааллаар нь үгийн бүтцээр зөв задалсан эсэхийг шалгадаг [6].

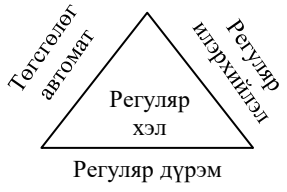
2.2 Төгсгөлөг төлөвт үг зүй

1956 онд Клиний анх боловсруулсан регуляр илэрхийлэл нь хэлэн дэх тэмдэгт мөрүүдийг (string) тодорхойлох томъёо болжээ [10]. Регуляр илэрхийлэлд ашигладаг үндсэн операторуудыг Хүснэгт 8-д харуулав.

Хүснэгт 8. Регуляр илэрхийллийн операторууд

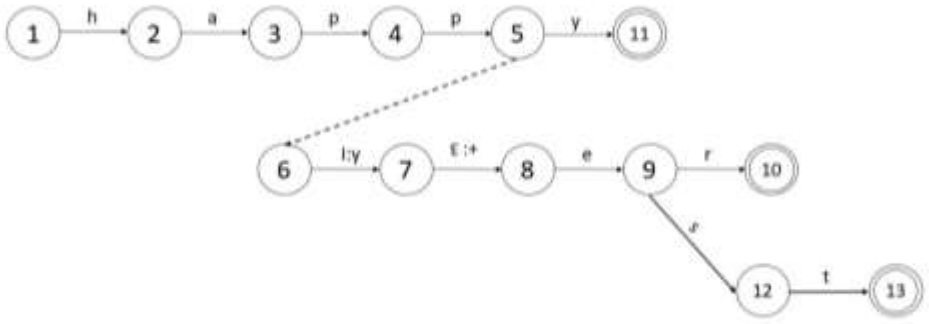
Регуляр илэрхийлэл	Нийцэлт
woodchuks	woodchuks
[wW]oodchuks	Woodchuks эсвэл woodchuks
[abc]	a эсвэл b эсвэл c
[А-Я]	том үсэг
[0-9]	нэг оронтой тоо
[^Ss]	S болон s-ээс бусад тэмдэгт
[^А-Я]	том үсгээс бусад тэмдэгт
colou?r	color эсвэл colour
beg.n	beg ба n хооронд ямар нэгэн тэмдэгт байна. Жнь: begin begun
a*	Клиний од. Тэг эсвэл түүнээс урт ямар нэгэн тэмдэгт мөр.
a+	Клиний нэмэх. Тэгээс урт ямар нэгэн тэмдэгт мөр.
cat:|(dog	дизьюнкци. cat эсвэл dog
Gupp(y|ies)	guppy эсвэл guppies

Регуляр илэрхийлэл нь төгсгөлөг төлөвт автоматыг тайлбарлах нэг арга юм. Регуляр илэрхийлэл бүрийг төгсгөлөг төлөвт автоматаар хэрэгжүүлж болно. Нөгөө талаас төгсгөлөг төлөвт автомат бүр регуляр илэрхийллээр тайлбарлагдаж болно.



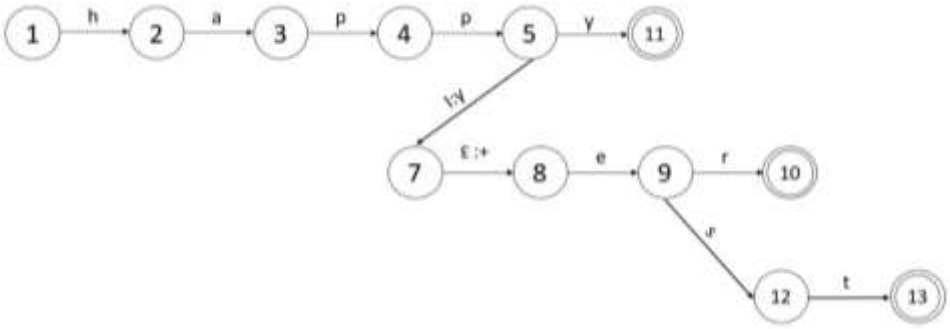
Зураг 8. Регуляр хэлийг тайлбарладаг тэнцүү чанартай аргууд

Төгсгөлөг төлөвт автоматаар оролтонд хувиргах үг, гаралтанд хувирсан үг гарах байдлаар үг хувилгах үйлдлийг гүйцэтгэж болно. Жишээ нь: англи хэлний happy гэдэг үг байхад түүнд happier гэсэн үг гарах happy+er буюу happy гэсэн үгэнд er нөхцөлийн хэлбэр залгах, happiest гэсэн үг гарах happy+est нөхцөл залгах төгсгөлөг автоматыг Зураг 9-д үзүүлэв.



Зураг 9. "happy" үгийн хувиллуудын төгсгөлөг төлөвт хувиргагч

Зураг 9-д 1 төлвөөс 2 төлөв хоорондох h нь h үсэг ороход гаралтын функц нь h үсэг гаргана гэсэн үг. Харин 6 төлвөөс 7 төлөв хоорондох i:y гэдэг нь y үсгийн оролтонд i үсэг гарна гэсэн үг, ɛ:+ гэдэг нь нөхцөл залгах гэсэн утгатай + тэмдгийн оролтонд ɛ буюу хоосон тэмдэгт гарна гэсэн үг. Мөн 5, 6 төлөв хоорондох үсэг байхгүй тасралттай зураас нь шууд шилжилтийг илэрхийлнэ. Шууд шилжилтийг Зураг 10-т харуулсан байдлаар хасаж илэрхийлж болно.



Зураг 10. "happy" үгийн хувиллуудын шууд шилжилтгүй төгсгөлөг төлөвт хувиргагч

Зураг 10-т 5 төлөвт байхад у оролтонд 11 төлөв рүү, 7 төлөв рүү шилжиж болно. Хэрэв 11 төлөвт байхад оролтын тэмдэгт мөр эцэстээ хүрээгүй бол буруу төлөв рүүгээ шилжсэн байна гээд 7 төлөв рүү шилжинэ. Иймээс дээрх зурагт эцсийн оролтын тэмдэгт ороход 11, 10, 13 буюу эцсийн (accept) төлөвт очсон бол зөв оролтын тэмдэгтүүд ба зөв гаралтын тэмдэгтүүд гэсэн үг. Хэрэв эцсийн оролтын дараа бусад (reject) төлөвт өчвол буруу оролтын тэмдэгтүүд байсан ба гаралтын тэмдэгтүүд нь мөн буруу гэсэн үг.

Иймээс төлвүүдийг дотоод төлөв, эцсийн төлөв гэж хоёр ангилна. Иймээс дээрх төгсгөлөг төлөвт автоматыг (A,Q,V,δ,q0,λ,F) томъёолж болно. Үүнд:

$\delta : Q \times A \rightarrow \text{set } Q$

$\lambda : Q \times Q \rightarrow V$ байна.

Үг зүй нь үгийг хувилгах, түүний эсрэг хувилсан үгийг бүтцээр задлах гэсэн хоёр үйлдэлтэй байдаг. Өөрөөр хэлбэл *happier* гэдэг үгийг *happy+er* болгон үгийн бүтцээр задлах шаардлагатай байдаг. Төгсгөлөг автомат нь нэг автоматаар энэ хоёр үйлдлийг гүйцэтгэх чадвартай байдаг. Ингэхдээ оролт, гаралтыг солиход хангалттай. Дээрх Зураг 10-т i:y байхад у нь оролт болж байсан бол одоо i нь оролт болно гэсэн үг. Kimmo Koskenniemi энэ санааг ашиглан үг зүйн хоёр түвшинт загвар гаргасан нь одоо олон орны бичгийн үг зүйд ашиглагдаж байна. Мөн Kimmo Koskenniemi үг хувирах зөв бичгийн

дүрмийг илэрхийлэх хоёр түвшинт дүрэм зохиож түүнээсээ төгсгөлөг автоматад хувирган ашигладаг [10].

(A,Q,V,δ,q0,λ,F) автомат ашиглах үг зүйн алгоритм дараах хэлбэртэй байна.

```
FSA (q, input, output)
      if  |input| > 0 then
            set ← δ(q, input[1])
            foreach s in Set
                  FSA(s, input[2...], output+λ(q,s))
            end  foreach
            else if q ∈ F then
                  return output
      end if
```

Энэ алгоритм нь төгсгөлөг төлөвт автоматын эхлэлийн төлвийг q параметрт өгч, оролтын тэмдэгт мөрийг input параметрт өгөхөд output буцах утганд гаралтын тэмдэгт мөр гарна. Хэрэв оролт нь *happier* байхад гаралт нь *happy+er* байна.

Төгсгөлөг төлөвт автомат ашиглан *үг зүйн задлан шинжилгээ* (morphological parsing) хийдэг алгоритмыг *төгсгөлөг төлөвт хувиргагч* (finite-state transducer) гэнэ. Төгсгөлөг төлөвт хувиргагч бол яриа болон хэлний боловсруулалтын чухал технологийн нэг юм.

Төгсгөлөг төлөвт хувиргагчийн оролт нь cats байхад гаралт нь cat+N+P1 буюу cat-ийн олон тоо cats байна.

Үг зүйн задлан шинжлэгчийг бий болгохын тулд ядаж дараах зүйлс хэрэгтэй [10]. Үүнд:

1. Үгийн сан
2. Морфотактикууд
3. Зөв бичгийн дүрмүүд

Үг зүйн задлан шинжилгээ

Төгсгөлөг төлөвт үг зүйн загварт үгийн харилцан зохицсон хоёр хэлбэр байдаг. Дүрмийн хэлбэр (lexical level) нь бүтээврүүдийн нийлбэр нь үгийг бүтээж байгааг илэрхийлнэ. Харин бичигдэх хэлбэр (surface level) нь бүтээврүүд зөв бичгийн дүрмийн дагуу үг бүтээж байгааг илэрхийлнэ. Зураг 11-д cats үгийн хоёр хэлбэрийг харуулав.

дүрмийн хэлбэр	c	a	t	+N	+P1
бичигдэх хэлбэр	c	a	t	s	

Зураг 11. Үгийн дүрмийн ба бичигдэх хэлбэрийн жишээ

Дүрмийн хэлбэрийн үг нь Σ цагаан толгойн тэмдэгтүүдээс тогтоно. Харин бичигдэх хэлбэрийн үг нь Δ цагаан толгойн тэмдэгтүүдээс тогтоно. Коскеннимийн хоёр түвшинт үг зүйн хувьд төлвийн шилжилт бүрд цагаан толгой бүрээс нэг тэмдэгт байхыг зөвшөөрдөг. Үүнийг оролт гаралтын буюу *боломжит хос* гэнэ. Зураг 11-д байгаа төлвийн шилжилт бүр боломжит хос болно. Жишээлбэл i:y ба ε:+. Хэрэв a:a байвал *өгөгдмөл хос* гэх ба шилжилтийг ганц a тэмдэгтээр товчилж дүрсэлж болно. Мөн ^ тэмдэгтээр *бүтээврийн эхлэлийг,* # тэмдэгтээр *үгийн төгсгөлийг* илэрхийлдэг.

Үгэнд нөхцөл залгахад нөхцөлийн ямар хэлбэр залгах, зөв бичгийн дүрмээр яаж хувилахыг тодорхойлохын тулд завсрын хэлбэр (intermediate level) тодорхойлдог. Зураг

12 fox+N+P1 дүрмийн хэлбэрийн үгийн хувьд завсрын хэлбэр болон бичигдэх хэлбэрүүдийг харуулав.

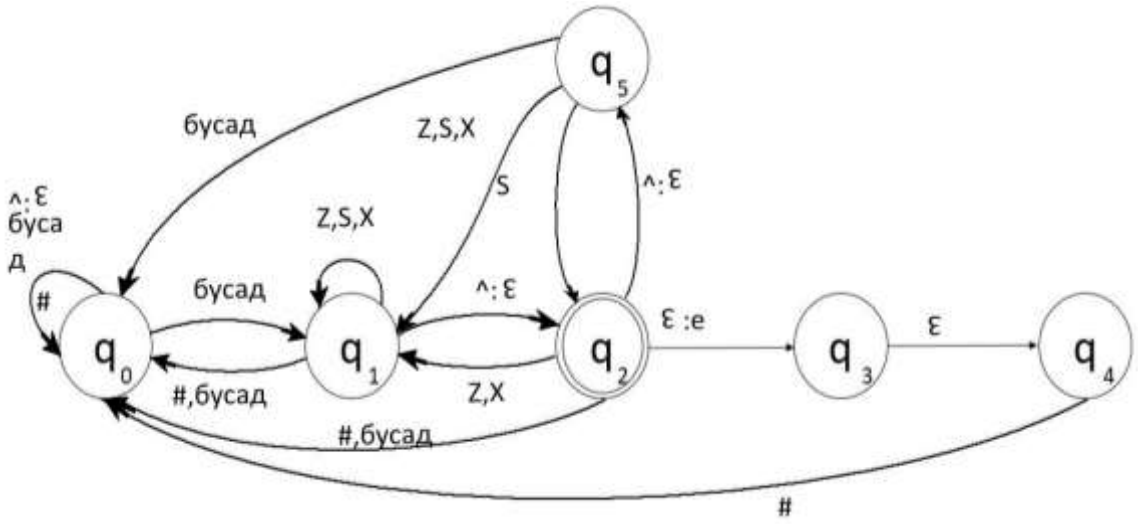
дүрмийн хэлбэр	f	o	x	+N	+P1	
завсрын хэлбэр	f	o	x	^	s	#
бичигдэх хэлбэр	f	o	x	e	s	

Зураг 12. Үгийн дүрмийн, завсрын ба бичигдэх хэлбэрийн жишээ

Англи хэлний "x, s, z-ээр төгссөн үгэнд s бүтээвэр залгахад e үсэг жийрэглэнэ" гэдэг дүрмийг Чоймскийн ба Halle нарын дүрмийн хийсвэрлэлээр илэрхийлбэл дараах байдалтай болно.

Үүнд: $$\varepsilon \rightarrow e / \left\{ \begin{matrix} x \\ s \\ z \end{matrix} \right\} \wedge \_ S\#$$ (2.1)

Энэ дүрэмд *a→b/c_d.* хэлбэр нь c ба d хоёрыш хоорондох a нь b болно гэдгийг заана. Мөн ε шууд шилжилт буюу хоосон тэмдэгтээр ашиглагдана. (2.1) дүрмийн автоматыг Зураг 2.6-д харуулав.



Зураг 13. (2.1) дүрмийн хувиргагч

Зураг 13-д байгаа "бусад" гэдэг нь автоматад байхгүй бусад тэмдэгтүүд ба хослолуудыг төлөөлнө. (2.1) дүрмийн шилжилтийн хүснэгтийг Хүснэгт 9-д үзүүлэв.

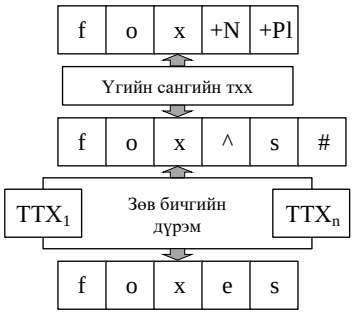
Хүснэгт 9. (2.1) дүрмийн шилжилтийн хүснэгт

төлөв\оролт	s:s	x:x	z:z	^:ε	ε:e	#	бусад
q_0:	1	1	1	0	-	0	0
q_1:	1	1	1	2	-	0	0
q_2:	5	1	1	0	3	0	0
q_3:	4	-	-	-	-	-	-
q_4:	-	-	-	-	-	0	-
q_5:	1	1	1	2	-	-	0

Төгсгөлөг төлөвт үгийн сан болон дүрмүүдийн нэгдэл

Үгийн сан ба шинжилгээ хийх, үгийн хэлбэр үүсгэх дүрмийн хувиргагч бэлэн болоход тэдгээрийг нэгтгэн үг зүйн систем бий болгоно. Зураг 14-д хоёр-түвшинт үг зүйн системийн архитектурыг харуулав.

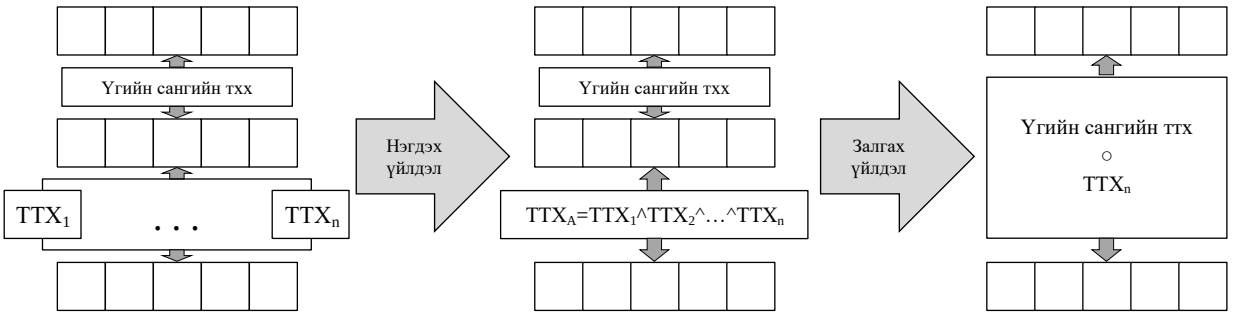
Үгийн сангийн хувиргагч нь үндэс ба бүтээвэр бүхий дүрмийн хэлбэрээс үндэс, бүтээврийн хэлбэрүүдийн нийлц бүхий завсрын хэлбэр лүү буулгана. Харин зөв бичгийн дүрэм бүрийг илэрхийлэх хувиргагчууд нь зэрэгцээ ажиллаж завсрын ба бичигдэх хэлбэрийн хооронд буулгана.



Зураг 14. Үгийн сангийн болон дүрмийн сангийн төгсгөлөг төлөвт хувиргагчид

Зураг 14-д харуулсан архитектурыг мөн хоёр-түвшинт хувиргагчийн цуваа гэнэ. Хувиргагчийн цуваа гэдэг нь эхний хувиргагчийн гаралт нь хоёр дахь хувиргагчийн оролт болно гэсэн үг юм.

Нэгдсэн төгсгөлөг төлөвт хувиргагч болгоход автоматын нэгдэх, залгах гэсэн хоёр үйлдэл хийгддэг (Зураг 15) [11]. Нэгдэх үйлдлийн математик буулгалтыг Каплан, Кэй нар 1994 онд тодорхойлж, Антров нэгтгэх алгоритмыг боловсруулсан. 1997 онд Мохри хувиргагчийг минимум хэлбэрт буулгах алгоритмыг тодорхойлсон [10].



Зураг 15. Хувиргагчуудын нэгдэх, залгах үйлдэл

Нэгдэх үйлдлийг ^, залгах үйлдлийг ○ тэмдгээр тэмдэглэнэ. Нэгдэх үйлдлээр хоёр автоматын аль нэг нь ажиллах боломжтой болдог. Харин O_1 гэсэн оролтгой $Г_1$ гэсэн гаралттай A_1 автоматын араас $Г_1$ гэсэн оролттой $Г_2$ гэсэн гаралттай A_2 автоматыг залгах үед залгагдсан автомат $A_1 \circ A_2$ нь O_1 гэсэн оролттой $Г_2$ гаралттай автомат үүснэ.

Үг зүйн шинжилгээний үед *хоёрдмол утгын* (ambiguity) асуудал гарах боломжгүй. Жишээлбэл foxes үгийг fox+V+3Sg ба fox+N+P1 гэж хоёр хэлбэрээр задлах боломжтой. Үүний аль хэлбэр байх ёстойг хувиргагч шийдэх боломжгүй. Тухайн үгийн орчин тойрны үгсийн тусламжтайгаар *үг зүйн утга сонгогчоор* (morphological disambiguation) үүнийг тодорхойлно.

Хоёр түвшинт үг зүй

Хэл шинжлэлийн онолын хувьд хоёр түвшинт үг зүйн хамгийн чухал тал нь үүсгүүр үг зүйн дахин бичих дүрмийн оронд үгийн сангийн хэлбэр болон өнгөн хэлбэр

гэсэн хос гишүүдийн хоорондын холбоог тайлбарласан дүрмүүдийн тогтвортой системийг оруулж ирсэн үндсэн өөрчлөлт болсон.

1983 онд Финляндын эрдэмтэн Киммо Коскенними тооцоолох хэл шинжлэлд хоёр түвшинт үг зүй гэсэн ойлголтыг оруулж ирсэн. Гол санаа нь үгсийг үг зүйн талаас нь авч үзэхдээ үгийн сангийн төвшин, өнгөн төвшин гэж тусад нь авч үзсэн. Тэрээр **surface form** буюу хувилсан хэлбэрийг **(S)**, **lexical form** буюу үгийн сангийн хэлбэрийг **(L)**[1] гэж тэмдэглээд үүсгүүр хэл зүйн дахиж бичих дүрмийг ашиглахгүйгээр шууд хооронд нь харьцуулж авч үзсэн.

(L) baatar+aas
(S) baat0r0aas

Үгийн сангийн болон өнгөн хэлбэр дэх үсэг болгоны хоорондох холбоог төгсгөлөг автоматыг ашиглан тодорхойлох боломжтой. Энэхүү системийн анхны хэлбэрийг Фин хэлийг шинжлэхэд амжилттай ашигласан. Түүнээс хойш янз бүрийн хэлэнд өргөн хүрээнд хэрэглэгдэх болсон. Монгол болон Фин хэл нь нийтлэг талууд ихтэй учраас (олон тооны дагаваруудтай “хоёр хэл хоёулаа залгамал хэл”, эгшиг зохицох ёс гэх мэт) Коскеннимийн загвар нь Монгол хэлний асуудлуудад тохиромжтой байна гэж таамаглаж болох юм.

Хоёр түвшинт дүрмийн компьютерын программ нь хоёр үндсэн үүрэг гүйцэтгэнэ. Үгийн өнгөн хэлбэрийг өгөхөд түүнд тохирох үгийн сангийн хэлбэрийг таних боломжтой ба үгийн сангийн хэлбэрийг өгөхөд түүнд тохирох өнгөн хэлбэрийг үүсгэх боломжтой байх ёстой.

Kimmo Koskenniemi "Two-level model of morphological analysis" нэртэй докторын (Ph.D) ажилдаа анх энэ загварыг тодорхойлсон байна [11], [12], [13], [14]. Үг зүйн хоёр-түвшинт загвар нь үгийн хэлбэр үүсгэх, үгийн бүтцийг шинжлэх хоёр үйлдэлд нэг төгсгөлөг төлөвт автомат ашигладаг юм [15]. Ингэхдээ үг нь дүрмийн (жишээ нь: морь + д) болон бичигдэх (жишээ нь: моринд) гэсэн 2 хэлбэртэй байна гэж үзэж, үгийн хэлбэр үүсгэх нь үгийн дүрмийн хэлбэрийг төгсгөлөг төлөвт автоматад оруулахад үгийн бичигдэх хэлбэрт, үгийн бүтцийг шинжлэх нь үгийн бичигдэх хэлбэрийг төгсгөлөг төлөвт автоматад оруулахад, үгийн дүрмийн хэлбэрт шилжих үйлдэл гэж тодорхойлжээ.

Төгсгөлөг төлөвт автомат нь хоёр-түвшинт дүрэм хэлбэрт бичсэн зөв бичгийн дүрмүүдийг (хоёр-түвшинт дүрмийн хөрвүүлэгч гэх) тусгай программаар хөрвүүлж гаргадаг. Жишээлбэл : Монгол хэлний зөөлний тэмдгээр төгссөн үгэнд заахын тийн ялгалын "д" нөхцөл залгахад зөөлний тэмдэг "и" болж "н" үсэг жийрэглэнэ гэсэн зөв бичгийн дүрмийг нь хоёр-түвшинт дүрмээр бичвэл:

ь : и <=> _ + : н д Жишээ үг : морь+д/моринд

Үүнд:
+ тэмдэг нь нөхцөлөөр хувилбал гэсэн үг.
<=> _ гэдэг нь "ь" үсгийн зүүн тал өөрчлөгдөхгүй гэсэн үг.
"д" гэдэг нь өөрчлөгдөхгүй гэсэн үг.

[1] *Surface form* болон *lexical form* нь хоёр түвшинт дүрмүүдийн хэвшмэл тэмдэглэгээ болно. Манай тохиолдолд 0 буюу тэг нь түүний оронд юу ч харагдахгүй гэдгийг илэрхийлэх бөгөөд + нь бүтээврүүдийн заагийг, V нь гол авиалбарын эгшиг болно. Ийм учраас үр дүнгийн гадаад хэлбэр нь baatraas болно.

Жишээлбэл в:ь <=> _+: н д дүрмийг төгсгөлөг төлөвт автомат хэлбэрт хөрвүүлбэл:

	ь	ь	+	д	=	үгийн дүрмийн хэлбэрт байгаа тэмдэгт
	и	ь	н	д	=	үгийн бичигдэх хэлбэрт байгаа тэмдэгт
төлөв 1	2	4	1	1	1	хэвийн төлөв
төлөв 2	0	0	3	0	0	"+" тэмдэг зөвшөөрөх төлөв
төлөв 3	0	0	0	1	0	"д" үсэг зөвшөөрөх төлөв
төлөв 4	2	4	5	1	1	"+" тэмдэг хорих төлөв
төлөв 5	2	4	1	0	1	"д" үсэг хорих төлөв

Үүнд: "=" тэмдэг нь бусад тэмдэгт гэсэн үг

Үгийн хэлбэр үүсгэх: төгсгөлөг төлөвт автоматад үгийн дүрмийн хэлбэрт байгаа тэмдэгт мөр оруулахад үгийн бичигдэх хэлбэрт байгаа тэмдэгт мөр гарна. Жишээ нь : үгийн дүрмийн хэлбэрт байгаа "морь + д" гэсэн өгөгдөл ороход үгийн дүрмийн хэлбэрт байгаа "ь" тэмдэгт хоёр байгаа тул хоёр үр дүн гарна. Автоматын эхний төлөв 1 байх ба шилжих алхам бүрд төлөв өөрчлөгдөх ба гаралт нь бичигдэх хэлбэрт байгаа тэмдэгт байна.

Оролт		м	о	р	ь	+	д
Төлөв	1	1	1	1	2	3	1
Гаралт		м	о	р	и	н	д
Алхам		1	2	3	4	5	6

Үүнд төгсгөлийн төлөв 1 байгаа тул зөв үр дүн байна.

Оролт		м	о	р	ь	+	д
Төлөв	1	1	1	1	4	5	0
Гаралт		м	о	р	ь	н	д
Алхам		1	2	3	4	5	6

Үүнд төгсгөлийн төлөв 0 байгаа тул буруу үр дүн байна.

Үгийн бүтцийг шинжлэх: төгсгөлөг төлөвт автоматад үгийн бичигдэх хэлбэрт байгаа тэмдэгт мөр оруулахад үгийн дүрмийн хэлбэрт байгаа тэмдэгт мөр гарна. Жишээ нь: үгийн бичигдэх хэлбэрт байгаа "моринд" гэсэн өгөгдөл ороход нэг үр дүн гарна. Автоматын эхний төлөв 1 байх ба шилжих алхам бүрд төлөв өөрчлөгдөх ба гаралт нь дүрмийн хэлбэрт байгаа тэмдэгт байна.

Үр дүн: 1.

Оролт		м	о	р	и	н	д
Төлөв	1	1	1	1	2	3	1
Гаралт		м	о	р	ь	+	д
Алхам		1	2	3	4	5	6

Үүнд төгсгөлийн төлөв 1 байгаа тул зөв үр дүн байна.

Хоёр-түвшинт дүрэм нь дараах хэсгээс бүтнэ. Үүнд:

БоломжитХос **Оператор** ЗүүнОрчин_БаруунОрчин

Боломжит хос нь L:S хэлбэртэй байна. L бол дүрмийн хэлбэрийн үсэг эсвэл тэмдэгт байна. S бол бичигдэх хэлбэрийн үсэг эсвэл тэмдэгт байна. Дүрмийн хэлбэрээс бичигдэх хэлбэр чиглэлтэй тодорхойлдог. Орчин нь боломжит хосын дагуу хувирах орчны баруун, зүүн орчныг тодорхойлж өгнө. Зүүн баруун орчин байж болно. Эсвэл аль нэг нь байж болно. Хэрэв үгийн төгсгөлд хувирал явагдах бол зөвхөн зүүн орчин байна. 4 төрлийн

операторооc хамаарах 4 төрлийн хоёр-түвшинт дүрэм байх ба эдгээр дүрмүүдийн ялгааг Хүснэгт 10-д харуулав.

Хүснэгт 10. Хоёр-түвшинт дүрмүүдийн ялгаа

	L:S⇒LC_RC	L:S⇐LC_RC	L:S⇔LC_RC	L:S/⇐LC_RC
Энэ орчинд L:S хувирлыг зөвшөөрөх үү ?	Тийм	Тийм	Тийм	Үгүй
Энэ орчинд зөвхөн L:S хувирлыг зөвшөөрөх үү ?	Тийм	Үгүй	Тийм	-
Энэ орчинд L нь үргэлж S болж хувирах ёстой юу ?	Үгүй	Тийм	Тийм	-

Хоёр түвшинт үг зүй ба Монгол хэл

Монгол хэлний хэл зүйн зорилгуудыг авч үзэхдээ хоёр түвшинт үг зүйг оруулж ирснээр бид дараах үндсэн асуудлуудыг хөгжүүлэх боломжтой.

- Үгийн алдаа шалгах систем

Үгийн алдаа шалгах системийн хувьд систем өгөгдлийн урсгалаас оролтыг хүлээн авч үгийн язгуурууд болон бүтээвэр зүйн мэдээллүүдээс бүрдэх өгөгдлийн сан дотроос тухайн оролтод тохирох үгстэй харьцуулна. Таних төлөвт ажиллахдаа систем **hel^ye** гэсэн оролтын хэлбэрийг хүлээн авах боловч гэсэн **helye** гэсэн оролт орж ирэхэд түүний өнгөн буюу хувилсан хэлбэрийг авч үгийн сангийн хэлбэртэй нь харьцуулах замаар энэ оролтоос татгалзана.

(L) hel+ye
(S) hel^ye

Өгөгдлийн сан нь гэсэн **hel^ye** тэмдэгтийг агуулахгүй, зөвхөн үгийн язгуурууд, дагавар, нөхцлүүд, хамгийн гол нь Монгол хэлний бүтээвэр авиалбар зүйн тодорхойлолтыг (жишээлбэл манай тохиолдолд **hel** гэсэн үндсэнд **+ye** нөхцөл залгахад яагаад **helye** биш **hel^ye** гэж бичих ёстойг тайлбарласан) агуулна.

- Леммчлэх систем (Lemmatization system)

Леммчлэх систем нь өмнөхтэй ижил механизмаар ажиллах боловч гаралт нь үгийн бүх бүтээврүүдийн хэлзүйн шинж чанаруудын талаархи нэмэлт мэдээллийг агуулдаг. **hel^ye** гэсэн тохиолдолд, цааш задрахгүй **hel** гэсэн үйл үгийн захирах хүсэх төлөвийн хэлбэр болохыг мэдэгдэнэ. Мөн **saal^chin** гэх мэтийн үгийг **saal^** гэсэн язгуур, тухайн үйлийн эзнийг тодорхойлох **+chin** гэсэн дагаварт задалж болно.

- Монгол бичгийн хөрвүүлэгч

Бичиг хөрвүүлэх системд хуучин монгол бичгийн хэлбэрийг нь үгийн сангийн хэлбэр болгож, орчин цагийн монгол хэлний хэлбэрийг нь өнгөн хэлбэр болгож тодорхойлж болох юм. Бидний анхны жишээн дээр дараах дүйцлийг гаргаж болно.

(L) baγatur+asa
(S) ba0at0r0aas

Таних төлөвт систем орчин цагийн монгол хэлний үгийн хэлбэрийг хүлээн авч түүний монгол бичгийн хэлбэрийг мэдэгдэх ба үүсгэх төлөвт үгийн сангийн хэлбэр болох монгол бичгээр бичсэн үгийн өнгөн хэлбэр болгож орчин цагийн монгол хэлний үгийг бүтээнэ.

- Үгийн хэлбэр таних систем

Үгийн хэлбэр таних систем нь үндсэндээ леммчлэх системтэй адил боловч гаралт нь үгийн үндэс, язгуур руу чиглэгдэхгүй харин аль болох бүрэн хэмжээний дүрмийн тайлбарт чиглэгдэнэ.

- Хоёрдмол утгыг шийдвэрлэх

Хоёрдмол утгыг шийдвэрлэгч нь Монгол хэл дээр бичигдсэн тэмдэгтүүдийг оролтонд хүлээн авч энэхүү өнгөн хэлбэрийг бүтээж болох бүх боломжит үгийн сангийн хэлбэрүүдийг танина. Зөвхөн энэ мэдээллээр хоёрдмол утгыг шийдвэрлэхгүй ч боловсронгуй систем нь нэмэлт мэдээллийг санал болгох боломжтой. Жишээлбэл, асуултанд байгаа үг жинхэнэ нэр, тэмдэг нэрийн алин болох гэх мэт. Энэ мэдээлэл нь ихэвчлэн зөв сонголтыг тодорхойлоход хангалттай байдаг.

Хоёр түвшинт дүрмийн хэвшмэл тэмдэглэгээнүүд

Хоёр түвшинт дүрмийн хийсвэр тэмдэглэгээг гурван хэсэгт хувааж болно. Үүнд: Бүтээвэр авиалбарын боловсруулалтын тодорхойлолт, боловсруулалт явагдах газар болох орчны тодорхойлолт, боловсруулалт ба орчны тодорхойлолтыг холбосон хамаарлын оператор.

Бүтээвэр авиалбарын боловсруулалт

Бүтээвэр авиалбар зүйн боловсруулалт нь үгийн сангийн хэлбэр **(L)**, өнгөн хэлбэр **(S)** гэсэн хос тэмдгээр тодорхойлогдоно. Үгийн сангийн хэлбэр ба өнгөн хэлбэрийн хоорондох дүйцэл нь хоёр үндсэн хэлбэртэй байдаг гэж үзье. Эхний тохиолдолд үгийн сангийн авиалбар нь үр дүн дэх өнгөн авиалбараас ялгаатай байвал **p:b** гэж тэмдэглэх ба “[хэрэв … бол] **p** нь **b** болно” гэж уншина. Хоёр дахь тохиолдолд үгийн сангийн тэмдэг нь өнгөн хэлбэрийн ангид орно. Энэ нь бүтээвэр авиалбар ба үндсэн авиалбар дээр ихэвчлэн тохиолддог. Энэ тохиолдолд **B:b** and **B:w** гэж тэмдэглэх ба “Бүтээвэр авиалбар **B**-г нэг бол **b**, үгүй бол **w** гэж ойлгоно” гэж уншина.

Орчин

Орчин нь боловсруулалт явагдах байрлалыг заах бөгөөд зүүн гарын болон баруун гарын орчны тодорхойлолтыг сонгож болдог. Ядаж нэг тодорхойлолт байх ёстой ба хоёр талынх хоёулаа байсан ч болно.

Зүүн гарын орчны дүрмийн жишээ: **V:0 /<= #___**
Баруун гарын орчны дүрмийн жишээ: **V:0 <= ___ +:0 V V**
Хоёр талтай орчны дүрмийн жишээ: **+: ^ <=>[Vf|i] Ca (Ca)___ye**

Хамаарлын оператор

Боловсруулалт болон орчныг холбосон оператор нь дараах дөрвийн аль нэг нь байна.

1. => Хам сэдвийг хязгаарлах дүрэм (***context restriction rule***) нь нэг хам сэдвийн орчинд дүйцэл олдож байхад тэрхүү хам сэдэв дотор өөр дүйцэл олдох боломжтой гэдгийг заана. Энэ дүрмийг мөн ***only if rule*** гэж нэрлэдэг. Antworth энэхүү дүрмийг ***only but not always*** гэж нэрлэсэн.

   L:S => E
   L-ийг зөвхөн **E** орчин дотор **S** гэж ойлгоно.
   E буюу **E**-гээс өөр орчинд **L**-ийг **S** гэж ойлгож болохгүй.
   Хэрэв **L:S** байвал заавал **E** орчинд байх ёстой.
   E дотор **L:** **S** байхыг зөвшөөрнө.

   Жишээ нь: Зөөлний тэмдгийн дараа үндсэн эгшиг, эгшигт гийгүүлэгч, ирээдүй цагийн **–x** нөхцөл орвол зөөлний тэмдгийг сольж **и** болгоно.

   ^:i => __+:0 [V:0|Cv|h]

2. <= Өнгөн хэлбэрийн албадлагын дүрмийг (***surface coercion rule***) мөн ***always but not only*** гэж нэрлэдэг. Энэ нь дүйцэл болон хам сэдэв нь хоорондоо заавал холбоотой байна гэдгийг илтгэнэ. Нэг орчин өгөхөд дүйцэл заавал олдох боловч яг тэр дүйцлийг гаргах өөр орчин бас байна. Өөрөөр хэлбэл энэхүү дүйцлийг гаргах орчныг тодорхойлох дүрэм нь зөвхөн ганц байх албагүй.

   L:S <= E
   L-ийг **E** орчин дотор үргэлж **S** гэж ойлгоно.
   E дотор **L**-ийг **S** гэж ойлгож болохгүй.
   Хэрэв **L** нь **E** орчинд байвал **L:S** байх ёстой.
   E дотор **L:** **S** байхыг зөвшөөрнө.
   Өөр газар **L:S** байж болно.

   Жишээ нь: Эр үгэнд үйлт нэрийн ирээдүй цагийн **–x** нөхцөл залгахад өмнө нь **а** эгшиг жийрэглэж бичнэ.

   +:a <= Vf C (C) ___ h

3. <=> Нийлмэл дүрэм (***composite rule***) буюу ***if and only if rule*** дүрэм нь дүйцэл зөвхөн ганц хам сэдэв дотор олдох ба хам сэдэв нь тэр дүйцлийг шаардана гэдгийг илтгэнэ. Antworth энэ дүрмийг ***always and only*** гэж нэрлэсэн.

   L:S <=> E
   L-ийг зөвхөн **E** орчин дотор үргэлж **S** гэж ойлгоно.
   L:S => E ба **L:S <= E**
   E дотор **L:S** байх ёстой. Өөр газар ийм байж болохгүй.

   Жишээ нь: Эр үгэнд **я, ё** эгшиг өмнөх гийгүүлэгчээсээ саланги дуудагдахаар орвол өмнө нь хатуугийн тэмдгээр тусгаарлаж бичнэ.

   +:^^ <=> Vf C (C) __ [ya,yo]

4. /<=Үгүйсгэлийн дүрэм **(*negation rule*)** нь өгсөн хам сэдэв дотор бүхэлд нь тухайн дүйцэл орж болохгүй гэдгийг заана. Дүйцэл нь энд хэзээ ч орохгүй боловч өөр хам сэдэв дотор орохыг зөвшөөрнө. Энэ дүрэм нь PC-KIMMO-д орсон боловч Коскенними өөрийн бүтээлдээ энэ талаар дурдаагүй юм.

   L:S /<= E

E орчин дотор **L**-ийг хэзээ ч **S** гэж ойлгохгүй.
E дотор **L**-ийг **S** гэж ойлгож болохгүй.
Хэрэв **L** нь **E** орчинд байвал **L: S** байх ёстой.

Жишээ нь: Оноосон нэрийн төгсгөлийн гийгүүлэгчийн өмнөх балархай эгшгийг гээхгүй.

V:0 /<= #Cap V* C* ___ C +:0 VV

2.3 Үг зүй хэрэглүүр

Кирилл үсгийн дүрэм анх 1942 онд хэвлэгдэж [Дамдинсүрэн, 1942], 1946 болон 1983 онд дахин засварлан гаргасан байдаг.

Энэ дүрэм нь орчин цагийн монгол хэлний халх аялгууны байдлыг харгалзсан боловч монгол хэлний авианы байрлалын хуулийг бүрэн тусгаагүйн улмаас олон зохиомол дүрэмтэй, тэдгээр нь үгийн язгуур, үндэс, дагавар, нөхцөлийн бүтцийг бараг бүх тохиолдолд эвддэг. Бид хүчинтэй буй кирилл үсгийн дүрмийг мөрдсөн бөгөөд үүний дагууд боловсруулахад олон зүйл бэрхшээл тохиолдож байна.

Нэр үгийн бүлэг

Кирилл үсгийн дүрмээр нэрийн хувилах бүлэг 32. Эдгээр бүлэг нь эгшгийн зохицлоос хамааран дотроо 1-4 хүртэл хуваагдана. Дашрамд дурдахад, тухайн үг олон тоонд хэрхэн хувирах нь утгаас шууд хамаардаг. Тэгэхээр нэр үг бүрт ямар олон тооны залгавар авч болох хийгээд тэрхүү залгавар нь үндсийн ямар хэлбэрт залгагдахыг тэмдэглэж өгсөн.

Жишээ болгон эхний гурван бүлгийг толилуулъя:

1-р бүлэгт эгшигт гийгүүлэгчээр төгссөн, балархай эгшиггүй, үндсэнд “*н*” гардаггүй үгс багтана. Нэрийн нөхцөлийг эгшгийн зохицлыг харгалзан шууд залгана.

Жишээ: 1.1 *санал, цуврал* 1.2 *ном, бодрол*
1.3 *хөл, өмсгөл* 1.4 *бэр, дэглэм*

саналууд, саналын, саналд, саналыг, саналаас, саналаар, саналтай, санал руу, саналаа,

2-р бүлэгт “*н*” – ээр төгссөн, балархай эгшиггүй, үндсэнд нь “*н*” гардаггүй үгс багтана. Гол онцлог нь харьяалахын тийн ялгалын “*-ы, ий*” гэсэн нөхцөлийн хэлбэр авна.

3-р бүлэгт заримдаг гийгүүлэгчээр төгссөн, балархай эгшиггүй, үндсэнд “н” гардаггүй үгс хамаарна. Энэ бүлгийн үгэнд өгөх оршихын тийн ялгалын “-д” хэлбэрийг залгахад зохих эгшгийг жийрэглэнэ.

Үйл үгийн бүлэг

Кирилл үсгийн дүрмээр үйлийн хувилах бүлэг 16. Үгийн бүлэг нь мөн эгшгийн зохицлоос хамааран 4 хүртэл хуваагдана.

Мөн гурван жишээ:

1-р бүлэгт урт эгшгээр төгссөн үгс орох бөгөөд урт эгшгээр эхэлсэн нөхцөл залгахад “*г*” жийрэглэнэ.

Жишээ: 1.1 *асуу-, бай-* 1.2 *боо-, зохио-*
1.3 *ерөө-, нөө-* 1.4 *нээ-, хүлээ-*

2-р бүлэгт богино “*и*” эгшиг, туслах эгшгээр төгссөн үгс багтах ба урт эгшгээр эхэлсэн нөхцөл залгахад урт эгшгийн нэгийг хасаж бичнэ.
3-р бүлэгт “*м, н, л, в*”-ээс бусад буюу “*р, г*” эгшигт гийгүүлэгчээр төгссөн үгс орно. Дан гийгүүлэгчээс бүтсэн болон давхар гийгүүлэгчээр эхэлсэн нөхцөл залгахад эгшиг жийрэглэнэ.

Бид үг зүйн хэрэглүүр хөгжүүлэхдээ Ц.Дамдинсүрэнгийн кирилл үсгийн дүрмийг баримтлан, нэр үг хувилах 32, үйл үг хувилах 16 дүрмийг үндэс болгон Монгол хэлний кирилл үсгээр бичигдсэн бичвэрт үг зүйн задлан ялгал хийх программын хөгжүүлэлтийг хийж ажиллалаа.

Зураг 16. Үг зүйн хэрэглүүрийн дэлгэцийн агшин

Үг зүйн үүсгүүр

Үг зүйн хэрэглүүрээр кирилл үсгээр бичигдсэн үгийг хувилгахдаа үндэс дээр нөхцөлийг нэг л удаа залгахад бөгөөд **нэр үндэс 32**, **үйл үндэс 16 бүлэг** дүрмээс бүрдэнэ. Давхар нөхцөл залгах тохиолдолд нөхцөлийн кодоор сангаас дуудаж залгах бөгөөд үг хувилгах дүрэм нэг давхар нөхцөл залгахтай адил хэрэглэгдэнэ. Жишээ нь:

8-р бүлэгт богино “и”-ээр төгссөн үгс орно. Бусад бүлгээс ялгарах онцлог нь урт эгшгээр эхэлсэн бүх залгавар нөхцөлийг залгахад нөхцөлийнх нь эхний эгшгийг гээж бичих буюу программд ойлгуулснаар нөхцөлийн нэг эгшигтэй хувилбар (*-ар, эс, йн* г.м)-ыг залгана. Программд дараах байдлаар илэрхийлнэ.

```
private String grammar8(string _word, string _suffix, string _SUFFIX_CODE)
{
   switch (_SUFFIX_CODE)
   {
      case "NP":
      case "NC4":
      case "NC5":
      case "NX1":
         return _word + _suffix.Remove(0, 1);
      case "NC1":
         if (_suffix.Substring(0, 2) == "ын")
            return _word.Remove(_word.Length - 1, 1) + "ийн" + _suffix.Remove(0, 2);
         else
            return _word.Remove(_word.Length - 1, 1) + _suffix;
      case "NC3":
```

```
            if (_suffix.Substring(0, 2) == "ыг")
               return _word.Remove(_word.Length - 1, 1) + "ийг" + _suffix.Remove(0, 2);
            else
               return _word.Remove(_word.Length - 1, 1) + _suffix;

         default:
            return _word + _suffix;
      }
}
```

Монгол бичгийн дүрмээр нэрийн хувилах дүрэм **6 бүлэг**, үйлийн хувилах дүрам **6 бүлэг**. Дэвсгэрлэх ёс, эгшгийн зохицлоос хамааран 2 хуваагдана.

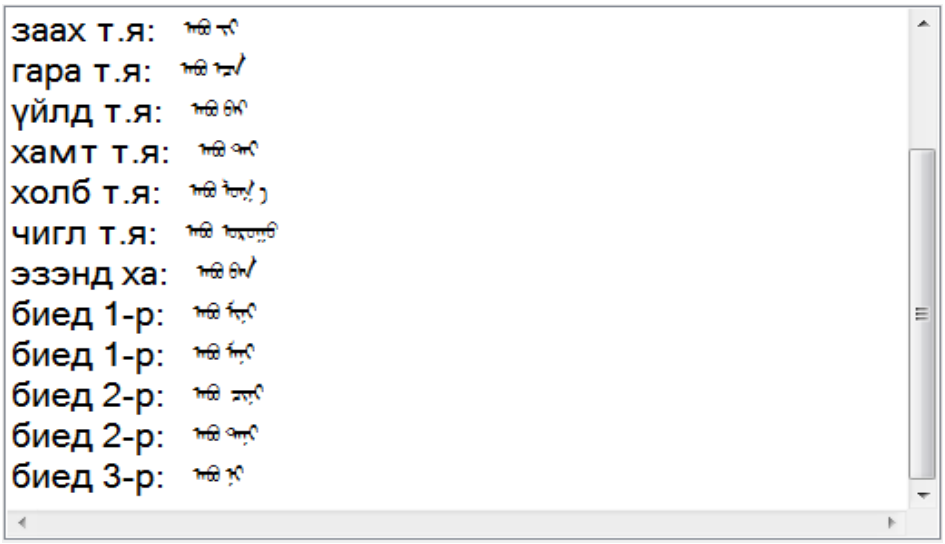
2-р бүлэгт эгшгээр төгссөн, тогтворгүй “*и*” бүхий үгс багтана.

```
private String grammarMonN2(String _word, String _suffix, string _SUFFIX_CODE)
{
    switch (_SUFFIX_CODE)
    {
        case "NC1":
        case "NC2":
        case "NC4":
             return _word + "ї" + _suffix;
        default:
             return _word + _suffix;
    }
}
```

Жишээ: Аав гэсэн нэр үгийг тийн ялгал, хамаатуулах нөхцөлөөр хувилгасан үр дүн.

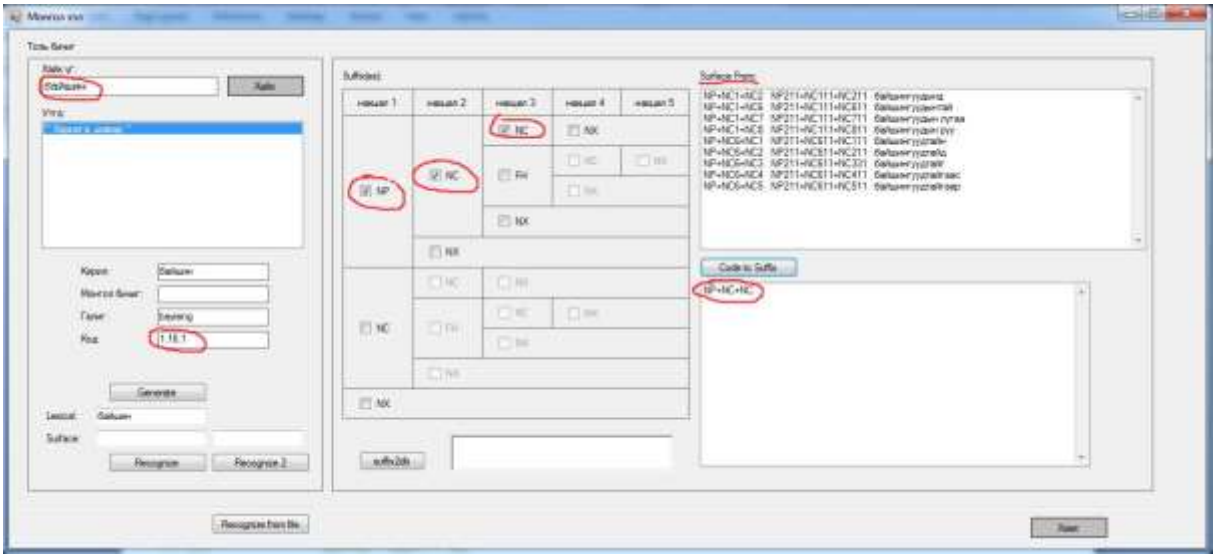
Олон тоо:	аавууд
Нэрлэх т.я:	аав
Харьяалах т.я:	аавын
Өгөх орших т.я:	аавд
Заах т.я:	аавыг
Гарах т.я:	аавaас
Үйлдэх т.я:	аавaар
Хамтрах т.я:	аавтай
Холбох т.я:	аав лугаа
Чиглэх т.я:	аав руу
Эзэнд хамаатуулах:	аавaа
Биед хамаатуулах (1-р):	аав минь
Биед хамаатуулах (1-р):	аав маань
Биед хамаатуулах (2-р):	аав чинь
Биед хамаатуулах (2-р):	аав тань
Биед хамаатуулах (3-р):	аав нь

Үг зүйн хэрэглүүрд "аав" гэсэн нэр үндсийг монгол бичгийн нөхцөлөөр хувилгасан байдал.



Зураг 17. Үг зүйн хэрэглүүрээр монгол бичгээрх үгийг хувилгасан байдал

Нэр үндсэнд 5 давхар нөхцөл давхарлан орох боломжтой. Программд энэ боломжийг нэмж оруулсан.



Зураг 18. Үг зүйн хэрэглүүрээр үгийг давхар нөхцөлөөр хувилгасан байдал

"байшин" гэсэн нэр үндсийг **олон тоо + тийн ялгал + тийн ялгал**ын нөхцөлөөр хувилгасан жишээг оруулав.

NP+NC1+NC2	NP211+NC111+NC211	байшингуудын
NP+NC1+NC2	NP211+NC111+NC611	байшингуудынтай
NP+NC1+NC2	NP211+NC111+NC711	байшингуудын лугаа
NP+NC1+NC2	NP211+NC111+NC811	байшингуудын руу
NP+NC1+NC2	NP211+NC611+NC111	байшингуудтайгийн
NP+NC1+NC2	NP211+NC611+NC211	байшингуудтайд
NP+NC1+NC2	NP211+NC611+NC331	байшингуудтайг
NP+NC1+NC2	NP211+NC611+NC411	байшинтайгаас
NP+NC1+NC2	NP211+NC611+NC511	байшинтайгаар

Бидний боловсруулсан Монгол хэлний үг зүйн загварчлалын MoMoTo хэрэглүүрийг (кирилл, монгол бичиг) үргэлжлүүлэн хөгжүүлж нэг үгэнд олон тооны нөхцөлийг үгийн утгаас хамааруулан зөв нөхцөл сонгодог болгон сайжруулсан.

Нөхцөлөөр хувилгах үндэс, үндсийн кодоос хамааруулан олон тооны нөхцөл сонгох кодын хэсгийг харуулав.

```
//----------------------------//
//   Олон тооны нөхцөл (NP)   //
//----------------------------//
private String suffixCodeNP(string _w, string _word_code, string _suffix_code)
{
   string suffix_code, B_Code, C_Code;
   string[] code = _word_code.Split('.');

   B_Code = code[1];
   C_Code = code[2];

   suffix_code = String.Format("{0}1{1}", _suffix_code, C_Code);

   return suffix_code;
}

//------------//
//   Нэр үг   //
//------------//
   case "NP":
   // Олон тоо
   string[] code = txtNP.Text.Split(',');
   if (code[0] == "N" | String.IsNullOrEmpty(code[0]))
      suffix_code_ext = SF_NULL;
   else
   {
      //foreach (string code_np in code)
      suffix_code_ext = suffixCodeNP(_LexicalForm, _WORD_CODE, code[0]);
   }
   break;
```

1.4a.4	шүүгдэгч	NP7	шүүгдэгчид
1.30.1	шуугиа	NP2	шуугианууд
1.2.1	шуугиан	NP2	шуугианууд
1.3.1	шуугиант		шуугиантууд
1.15.1	шуугиантай		шуугиантайнууд
1.5.1	шуугиур	NP2	шуугиурууд
1.5.4	шүүгүүр	NP2	шүүгүүрүүд
1.4a.4	шүүгч	NP1, NP7	шүүгч нар, шүүгчид
1.19a.4	шүүгчид	N	
1.12.4	шүүгээ	NP2	шүүгээнүүд
1.21.4	шүүгээнцэр		шүүгээнцрүүд
1.22.1	шуудаг	NP2	шуудгууд
1.15.1	шуудай	NP2	шуудайнууд
1.15.1	шуудай	NP2	шуудайнууд

Үг зүйн загварчлалын программд олон тооны нөхцөлийг зөв сонгодог болгохын тулд Монгол хэлний үгийн үндсийн хөмрөг дээр олон тооны нөхцөлийг хадаж баяжуулалт хийсэн.

Хүснэгт 11. Үг зүйн хэрэглүүрт ашиглаж буй сангийн загвар

Үгийн код	**Үндэс**	**Олон тооны нөхцөл**
1.20.1	алдалтан	NP2,NP7
1.17.1	алдам	
1.4.1	алдамч	NP1,NP7
1.8.1	алданги	NP2
1.21.1	алдар	NP2
1.3.1	алдарт	NP5,NP6
1.15.1	алдартай	NP5,NP6
1.20.1	алдартан	NP2
1.17.1	алдаршил	
1.15.1	алдаршингуй	
1.1.1	алдаршмал	
1.21.1	алдас	NP2
1.31.1	алдаш	NP2

Үг зүйн задлуур

Үгэнд үг зүйн задлал хийхэд кирилл бичгийн нэр үгийг **18 бүлэг**, үйл үгийг **31 бүлэг** дүрмээр задална. Жишээлбэл:

ь + и(Э)	(ботиуд = боть + ууд)
*(ЗГ)и → *(ЗГ)ь + (ЗГ)*	(ботид = боть + д, ботитой = боть + той)
и + и(Э)	(уушгиар = уушги + аар)
*ин → *ь + (ЭЭ)*	(мориноос = морь + оос)

Зураг 19. Үг зүйн хэрэглүүрээр үгэнд үг зүйн задлал хийсэн байдал

Хувилсан үгийг задлахдаа **үндэс + нөхцөл** гэсэн хоёр бүтээвэр болгоно.

Үг зүйн задлал зөв хийсэн эсэхийг буцаж шалгах функцийг оруулсан.

```
private bool checkRecognize(string w, string scode, string surface)
{
    List<string> recCodes = new List<string>();
    recCodes = getRootCode(w);
    string sufcode;
    sufcode = (scode.Length > 3) ? scode.Substring(0, 3) : scode;

    foreach (string c in recCodes)
    {
        if (getSurfaceForm(w, c, sufcode) == surface)
            return true;
    }

   return false;
}
```

Жишээлбэл. "ангийн" гэдэг үгийг үндэс нөхцөлөөр задлахад дараах үр дүн гарна.

3. анги + ийн -> ангийн ✓
ийн NC113
ийн NC114

4. ан + ийн -> ангийн ✓
ийн NC113
ийн NC114

6. анга + ийн -> ангын ✗
ийн NC113
ийн NC114

7. анаг + ийн -> анагийн ✗
ийн NC113
ийн NC114

Үгэнд үг зүйн задлал хийхэд монгол бичгийн нэр үгийг **2 бүлэг**, үйл үгийг **3 бүлэг** дүрмээр задална.

```
// *н → *    (nidun_u, nidun_du, nidun_aca)
if (w.Length > 1 && w[w.Length - 1] == 'ᠨ')
{
    suffix = s;
    root = w.Remove(w.Length - 1);

    if (CheckMonRS(root, "1") && CheckMonSF(suffix))
    {
        valid_ws.Add(String.Format("16. {0} + {1}    (-{2})", root, suffix, s_cyr));
        if (checkRecognize(root, s_code, txtSurface.Text))
            valid_ws.Add("✓");
        getSuffixCodeMon(valid_ws, suffix);
    }
}
```

Зураг 20. Монгол бичгээр бичигдсэн үгийг үг зүйн задлал хийсэн байдал

2.4 Бүлгийн дүгнэлт

Төслийн үр дүнгийн даалгаврын дагуу компьютерын тусламжтайгаар кирилл ба монгол бичгээр бичсэн үгэнд үг зүйн шинжилгээ хийх, үгийг зөв бичгийн дүрмийн дагуу нөхцөлөөр хувилгах аргыг тодорхойллоо. Энэхүү нийлмэл процесс нь компьютер хэл шинжлэлийн ухааны компьютерын үг зүйн салбарт хамаарах ба эх хэлийг компьютерээр боловсруулах эхний ажлуудын нэг билээ. Компьютерын үг зүйн шинжилгээ нь үгийн хэлбэр үүсгэх /generation/, үгийг бүтцээр задлах /analysis/ гэсэн үндсэн хоёр үйлдэлтэй.

Энэ бүлэгтээ монгол хэлний кирилл болон монгол бичгийн үгэнд үг зүйн боловсруулалт хийдэг өөрсдийн боловсруулсан үг зүйн загварчлалын хэрэглүүрийн тухай өгүүлсэн болно. Алгоритмын боловсруулж холбогдох программ хангамжийг бичиж, туршин үг зүйн загварчлалын хэрэглүүрийн бүтээх явцад гарч байсан асуудлууд, тэдгээрийг шийдвэрлэж байсан арга замуудад үндэслэн дараах дүгнэлтүүдийг хийж байна.

1. Монгол хэлний үг зүйд боловсруулалт хийхийн тулд юуны өмнө бусад олон хэлэнд амжилттай ашиглагдсан төгслөгөг автомат болон хоёр түвшинт үг зүйн загварыг эх хэлэндээ хэрэглэх боломжийг судлах байлаа. Компьютерын үг зүй (computational morphology)-н түвшинд үгийн хувиллыг таниулах буюу үгийг хувиллаар нь задлах болон үүсгэх хоёр чиглэлт үйлдлийг төгсгөллөг төлөвт автомат (finite state automata)-д үндэслэх нь үр дүнтэй гэж үзсэн. Тиймээс өгөгдлийн сангийн нэгж үгийг монгол хэлний үг бүтэх загварын дагуу хувилгах автоматыг дүрслэх нь тун чухал болно. Автомат нь дотроо хоёр төрөлтэй байдаг. “*Deterministic*” гэдэг нь нэг үр дүнг үүсгэхээр нэг төлөвөөс нөгөө төлөвт шилжихдээ зөвхөн ганц цагаан замаар яваад үүсгэж чадаж байх шинжийг хэлнэ. “*Nondeterministic*” гэдэг нь нэг үр дүнг нэгээс олон буюу салаа замаар үүсгэх боломжтой байх шинжийг хэлнэ. Бидний хувьд үр дүн бол “*үгийн хувилал*” юм. Иймд үр ашгийн хувьд “*deterministic*” нь илүү гэж үзсэн.
2. Монгол хэлний хувьд төгсгөлөг автоматыг *нэрийн* ба *үйлийн* гэсэн хоёр хувилбараар боловсруулах шаардлагатай. Энэ зорилгийнхоо хүрээнд төгсгөлөг автомат, хоёр түвшинт үг зүйн онолын үндэс болон монгол хэлний үг зүйг судалж монгол хэлний үг хувилах загварт тулгуурлан үг зүйн загварчлалын хэрэглүүр шинээр зохион монгол хэлний үг зүйд боловсруулалт хийн монгол хэлний үгсийг хэрхэн бүтээж, хэрхэн таньж бүтээврүүдэд задалж байгааг харуулсан. Үүний тулд хамгийн эхэнд дараах ажлуудыг хийж гүйцэтгэв.
   - Монгол хэлний үг зүйн хувилалын талаар судлан шаардлага хангахуйц санг Ц.Дамдинсүрэн, Б.Осор нарын “Монгол үсгийн дүрмийн толь” [3] хэмээх зөв бичих дүрмийн толинд орсон бүх идэвхтэй язгуурыг хамруулан үгийн өгөгдлийн сангийн файлыг кирилл болон монгол бичгээр бүрдүүлсэн.
   - Мөн дүрмийн файлд Монгол хэлний үг зүйн дүрмүүдийг кирилл болон уламжлалт монгол бичгийн хувьд тус тусад нь тодорхойлж оруулсан.
3. Зохиосон алгоритмын дагуу холбогдох программыг бичин шалгахад бидний боловсруулсан үг зүйн загварчлалын хэрэглүүрийг монгол хэлэнд хэрэглэж болох нь харагдаж амжилтай хэрэгжин сайн үр дүн авчирсан тул бид нийт байгуулсан сангаараа үгийн сангийн файлаа үүсгэн дараагийн шатны судалгаандаа ашиглахаар

шийдвэрлэсэн ба төгсгөлөг автоматыг ашигласан хоёр түвшинт үг зүйн загварыг хэл шинжлэлд хэрэглэж байгаа энэхүү судалгаа нь математик, компьютерын ухаан, хэл шинжлэлийн салбарыг холбосноороо онцлог юм.

4. Кирилл болон монгол бичгийн үг зүйн судалгаа хийж улмаар үг зүйн хэрэгсэл боловсруулахын өмнө энэ бүлэгт өгүүлсэн зүйлүүдийг зайлшгүй судлах шаардлага үүсч энэ бүлэгтээ компьютер хэл шинжлэл дэх үг зүй, үгийн утга таних болон үг зүйн загварчлалын талаарх судалгаануудыг авч үзсний үндсэн дээрээ дараах дүгнэлтүүдийг хийсэн.
   - үг зүйн шинжилгээнд үгийг бүтцээр задлах буюу recognition, үг хувилгах буюу generation гэсэн хоёр үндсэн үйлдэлтэй байдгаас үгийг бүтцээр *задлах* үйлдлийн оролт нь үг байна, гаралт нь үгийн үндэс болон залгасан нөхцөлийн дараалал байна. Харин үг хувилгах үйлдлийн оролт нь үгийн үндэс болон залгах нөхцөлийн дараалал байна, гаралт нь хувилсан үг байна.
   - Үг зүйн машин сургах аргууд нь одоохондоо тодорхой нэг хэлэнд дангаар нь шууд ашиглах хэмжээнд хүрээгүй байна гэж үзэж байна.
   - Үгийн утга автоматаар таних аргуудаас тогтвортой хэллэгт нэг утга гэсэн Яровскийн болон толь бичигт үндэслэсэн утга танхих аргуудыг монгол хэлэнд ашиглах нь зүйтэй гэж үзлээ. Эдгээр нь нэлээд өндөр (96%) нарийвчлалтай бөгөөд монгол хэлний зарим онцлог шинж чанарт таарах магадлал өндөр байна. Яровскийн боловсруулсан арга нь зөвхөн хоёр утгатай байхад ашиглаж байсан бол монгол хэлэнд хоёроос олон утгатай үг олон байдаг тул энэ алгоритмыг монгол хэлэнд хэрэглэж болохуйц болгох шаардлагатай.
   - Төгсгөлөг төлөвт автомат нь үг зүйн шинжилгээний хамгийн өргөн ашиглагддаг арга бөгөөд энэ аргаар нийт үг зүйн шинжилгээний 50 хувийг боловсруулжээ. Мөн энэ аргаар хамгийн олон хэлний үг зүйн шинжилгээг хийдэг бол үг зүйн хоёр түвшинт загвар нийт үг зүйн шинжилгээний 17 хувийг эзэлж байна.
   - Иймээс дээрх 2 аргыг ашиглан монгол хэлний үг зүйд шинжилгээ хийхэд тодорхой үр дүнд хүрнэ гэж дүгнэж үг зүйн загварчлалын хэрэглүүрийн алгоритмыг боловсруулан холбогдох программ хангамжийг бүтээж монгол хэлний үг зүйд шинжилгээ хийхээр шийдвэрлэсэн. Үүний тулд шаардлага хангасан монгол хэлний үгийн сангийн болон монгол хэлний дүрмүүдийг загварчилсан дүрмийн файл үүсгэн туршсан.

5. Ажлын явцад бидний боловсруулсан энэхүү хэрэгсэл нь монгол хэлний кирилл ба монгол бичгийн үг зүйн загварчлалд хэрэглэж болох нь тодорхой болсон ба энэхүү үг бүтээж, задалж байгаагаа дараагийн шатны судалгаандаа ашиглах бүрэн боломж нээгдлээ. Туршилтаас үзэхэд байгуулсан өгөгдлийн сан нь монгол хэл шинжлэлийн чиглэлээр хийх ажлын суурь болж чадахуйц зөв бүтэцтэй болсон байна. Үг хувилгах автоматын дагуу монгол хэлний үг зүйн шинжилгээг амжилттай хийж гүйцэтгэсэн ба үгийн утга таних сургалтын санг хангалттай хэмжээнд бүрдүүлж өгсөн нөхцөлд бидний сонгосон аргаар монгол үгийн утга таних боломжтой үр дүнг үзүүлэхээр байна.

6. Үгийн үндэс нь тухайн үгийн үндсэн утгыг хадгалах тул сургалтын жишээнээс үгийн үндсийг ялган хадгалах нь зүйтэй гэж үзсэн. Мөн кирилл болон монгол бичигт ижил

бичлэгтэй нэр ба үйл үг байх боломжтой тул сургалтын жишээнд нэр ба үйл үгийг заах нь илүү үр дүнтэй байна.

7. Монгол хэлний үг, өгүүлбэрийг компьютерын түвшинд таниулан ойлгуулж монгол хэлний зүй тогтол, өвөрмөц үзэгдлүүдийг орчин үеийн арга, технологийн туслалцаатайгаар нээн харуулж, судлах боломжийг бий болгох нь цаашдын судалгааны ажлын чухал үр дүн юм. Энэ нь уламжлалт хэл шинжлэлд тэр бүр илэрч харагддаггүй байсан хэлний онцлог үзэгдлүүдийг илрүүлж харуулах бололцоо олгож байгаад гол ач холбогдол нь оршино.
8. Өнөөдөр монгол хэлний материалыг ашигладаг төрөл бүрийн программ хангамжууд зохиогдож байгаа нь нэг үеэ бодвол сайшаалтай боловч монгол бичгийн хувьд тун цөөхөн байна.
9. Монгол хэлний үг зүйг загварчилж өгснөөр бид компьютер хэл шинжлэлийн бусад салбаруудыг хөгжүүлэх боломжтой гэж үзэж байна.
10. Бидний энэ судалгааны ажил нь монгол хэлний хэл шинжээчид, хэл шинжлэлийн салбарын оюутнуудад тус болох төдийгүй цаашид хийх шаардлагатай монгол хэлний өгүүлбэр зүйн задлагч, монгол бичгийн хөрвүүлэгч зэрэг программ хангамжийг хийж гүйцэтгэх суурь нь болно. Төгсгөлөг автоматыг ашигласан хоёр түвшинт үг зүйн загварыг хэл шинжлэлд хэрэглэж байгаа энэхүү судалгаа нь математик, компьютерын ухаан, хэл шинжлэлийн салбарыг холбосноороо онцлог юм. Ажлын явцад монгол хэлний үг зүйд хоёр түвшинт төгсгөлөг төлөвт үг зүйг хэрэглэж болох нь тодорхой болсон. Энэхүү үг бүтээж, задалж байгаагаа дараагийн шатны судалгаандаа ашиглах бүрэн боломж нээгдлээ.
11. Үг хувилгах автоматын дагуу монгол хэлний үг зүйн шинжилгээг амжилттай хийж гүйцэтгэлээ. Үгийн утга таних сургалтын санг хангалттай хэмжээнд бүрдүүлж өгсөн нөхцөлд бидний сонгосон аргаар монгол үгийн утга таних боломжтой үр дүнг үзүүлэхээр байна.

Ажлын үр дүнд кирилл болон монгол бичгээрх бичвэрт боловсруулалт хийх бүрэн боломжтой болсон бөгөөд үг зүйн загварчлалын программын анхны хувилбарыг боловсруулан бэлэн болгоод байна. Бичвэр боловсруулах ажлын үр дүн нь тэмдэгтийн кодлолтоос (латин, кирилл гэх мэт) хамааралгүй бөгөөд гол нь үгийн сангийн файлаа хэр хангалттай бүрдүүлж, зөв зүйтэй ангилав, дүрмээ хэр зөв тодорхойлов гэдгээс шууд хамаарч байна.

III. КИРИЛЛ БОЛОН МОНГОЛ БИЧГИЙН БИЧВЭРИЙН АЛДАА ИЛРҮҮЛЭХ, ЗАСАХ

1941 онд Монгол шинэ үсгийн цагаан толгойн бүрэлдэхүүн хийгээд түүний зөв бичих зүйн урьдчилсан дүрмийг боловсуурлснаас хойш хэрэглээний явцад засан сайжруулсаар ирсэн билээ. Энэ хооронд бага, дунд хэмжээний зөв бичих зүйн хэд хэдэн толь хэвлэгдэн гарсан бөгөөд хамгийн сүүлд өргөн олон нийт хийгээд бүх шатны сургалтын байгууллагын хэрэгцээнд зориулж идэвхтэй хэрэглэгддэг 18000 үгийг багтаасан (олон янз бичдэг үгсийг жигдлэн журамласан) үсгийн дүрмийн толийг Ц.Дамдинсүрэн, Б.Осор нар төрөөс өгсөн албан даалгаврын үндсэн дээр 1983 онд боловсруулан нийтэлж, бид өнөөг хүртэл даган мөрдөж байна.

1990-ээд оны үеэс манай улсад өрнөж эхэлсэн нийгмийн шилжилт, олон ургалч үзэл хандлагатай уялдан гарч ирсэн шинэ санаачилгууд ч бас монгол хэлний зөв бичих зүйн дүрмийг сайжруулах, шинэчлэх асуудлыг хөндөж байв. Гэвч өнөөг хүртэл иргэд болон эрдэмтдийн дундаас гарсан санаачилгууд нь үсгийн дүрмийн доторх зарим нэгэн гажилтыг арилгах эсхүл үсгийн дүрмийн тогтолцоог бүхлээр нь өөрчлөх, цагаан толгойн бүрэлэдхүүнийг цомхотгох зэрэг асуудалд чиглэж байсан учир хэл бичгийн салбарын эрдэмтэд хийгээд нийт иргэдийн дэмжлэгийг хүлээж чадаагүй юм. Энэ нь зөв бичих зүйн дүрэм байн дахин өөрчлөгдөөд буй мэт сэтгэгдэл төөрөгдлийг нийгэмд төрүүлсээр байгаа учир Хэлний бодлогын үндэсний зөвлөлөөс 2016 оны 9-р сарын 8-нд “Монгол үсгийн дүрэм батлах тухай” тогтоол гаргаж Ц.Дамдинсүрэн, Б.Осор нарын 1983 онд хэвлүүлсэн “Монгол үсгийн дүрмийн толь”-ийн хавсралт дахь “Монгол үсгийн дүрэм”-ийг баталгаажуулсан.

Нөгөө талаар дээрх толь бичиг хэвлэгдэн гарснаас хойших хугацаанд монгол хэлний үгийн сангийн идэвхтэй үгсийн тоонд олон мянган үг шинээр үүсэх болон гадаад хэлнээс нэвтрэн орж идээшсэн байна. Тиймээс дээр дурдсан хүчин төгөлдөр дүрмийг баримтлан Ц.Дамдинсүрэн, Б.Осор нарын “Монгол үсгийн дүрмийн толь”-ийг нэмэн баяжуулж тус улсын төр хувийн хэвшлийн байгууллагууд, нийт ард иргэдийн хэрэгцээнд нийлүүлэх, нийгэмд үүсээд буй үсгийн дүрэмтэй холбоотой эргэлзээ төөрөгдлийг арилгах шаардлага зүй ёсоор урган гарч байгаа билээ.

Мэдээллийн технологийн хэмээн нэрлэгдэж буй энэ зуунд ном сонин, хэвлэл, зурагт хуудас, албан бичиг зэрэг бүхий л бичиг баримтууд компьютер дээр боловсруулагдаж, хүн бүрийн өдөр тутмын хэрэглээнд нэвтэрч байна. Урьд нь хүмүүс зөвхөн мэдээллийг хүлээн авагч байсан бол өнөөдөр хүн бүр мэдээлэл түгээгч болсноор алдаа ихтэй материалууд түгээмэл хэвлэгдэх болсон. Үүнийг дагаад монгол хэл бичгийн дүрмийн алдаа ихээхэн газар авч түрэх хандлагатай болжээ.

Хүн бүр мэдээлэл түгээгч болсон өнөө үед монгол хэл бичгийн дүрмийн алдаа ихээхэн газар авч байна. Тодруулж хэлбэл сүүлийн жилүүдэд мессэж, чат, сошиал орчны нөлөөгөөр зөв бичих дүрмийн алдаа маш их гаргадаг болсон нь бодит үнэн билээ. Албан бичгийн харилцаанд зөв бичгийн дүрмийн алдаа гаргах нь байж болшгүй зүйл бөгөөд энэхүү таагүй байдлаас урьдчилан сэргийлэхийн тулд таны бичсэн текстийг хянаж, алдаа шаардлагатай байна. Бид энэхүү төрлийн программ хангамжийг вэбд суурилсан болон апликейшн хэлбэртэй зэргээр гаргах нь тун чухал байна.

3.1 Алдаа илрүүлэх, засах алгоритмын судалгаа

Дараах 4 төрлийн алдаа шалгах алгоритмын тухай судлаж туршсан.

1. Levenshtein Distance
2. N-gram
3. Jaro
4. SmithWaterman

Эхний 2 алгоритм нь үгийн алдааг олоход өргөн хэрэглэгддэг бол дараагийн хоёр алгоритм нь үгийн алдаа гэхээсээ илүүтэйгээр тухайн хэлээр нэрлэсэн нэгж таниурт хэрэглэх нь тохиромжтой байдаг. Иймд манай баг төслийнхөө хүрээнд Levenshtein Distance, N-gram гэсэн хоёр алгоритмыг ашиглаж үгийн утгын болон бичиглэлийн алдааг олоход ашигласан.

3.1.1 Левенштэйны алгоритм

Levenshtein Distance алгоритмын математик илэрхийлэл

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{хэрэв } \min(i,j) = 0, \\ \min = \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{бусад үед} \end{cases}$$

Үсэг нэмэх, хасах, эсвэл орлуулах (insertion, deletion and substitution - IDS): Нэг ба түүнээс дээш үсэг нэмж, хасаж, эсвэл орлуулж бичих гэсэн алдаануудыг математик загварчлалд оруулсан байдал юм.

Левенштэйний алгоритмыг 1965 онд Зөвлөлтийн математикч Владимир Левенштейний нэрээр нэрлэжээ. Өнөөдрийг хүртэл энэхүү алгоритмыг компьютер хэл шижлэлийн тэр дундаа үгийн алдаа шалгах (spell checker) программ хангамж хөгжүүлэхэд түгээмэл хэрэглэж ирсэн байна. Компьютерын хэл шинжлэлийн шинжлэх ухаанд Левенштэйн зай гэдэг нь хоёр дарааллын хоорондох ялгааг хэмжих мөрийн хэмжүүр юм. Товчхондоо хоёр тэмдэгт мөрийн ялгааг тодорхойлох алгоритм юм. Зөв бичих дүрэмд энэ алгоритмыг ашиглахдаа үгийн санд байгаа бүх үгтэй шалгах үгийг харьцуулдаг. Жишээ нь: “бичг” үгийг зөв эсэхийг үгийн сангийн “бичиг” харьцуулахад:

- “бичг” үг “бичиг” болоход хамгийн багадаа хэдэн өөрчлөлт орсон. Нэг өөрчлөлт гэдэгт үсэг гээгдэх, жийрэглэх, хувирах зэрэг орно.
- Хаана эдгээр өөрчлөлт орсон гэдгийг олно.

Хүснэгт 12. Левенштэйний алгоритмын жишээ

Алдаатай үг	Санал болгосон үгс	Тэмдэгтийн зөрүү
(бичг)	/бичиг/	1
	/бичгээ/	2
	 /ичиг/	2
	/бичдэг/	2

	ᠪᠢᠴᠢᠵ/ бичиж/	2

Хэрэв санд байгаа үгтэй ижилхэн байвал 0 гэсэн утга буцаана. Ижил биш буюу алдаатай тохиолдол 2 үгийн тэмдэгтийн зөрүүг буюу тоон утгыг буцаана. Хэрвээ санд байхгүй бол зөв үг ч байсан алдаатай гэж үзэх болно. Тийм учраас энэхүү алгоритм нь маш олон үгтэй өгөгдлийн сан дээр алдаа багатай ажиллана.

Хүснэгт 13. Левенштэйний алгоритмын сайжруулалт

№	Levenshtein	Damerau levenshtein	Үүрэг
1	word - worfd	word - worfd	Үсэг нэмэж бичих
2	word - wor	word - wor	Үсэг хасаж бичих
3	word - wold	word - wold	Үсэг сольж бичих
4		word - wodr	Үсгийн байрыг сольж бичих

Левенштэйний алгоритм нь үсэг нэмэх, хасах, өөр үсэг бичсэн байх гэсэн 3 төрлийн алдааг олж чаддаг байсан. Хэрвээ 2 үсгийн байрыг сольж бичсэн алдаа байвал яах вэ гэдгийг эх хэл боловсруулалтын анхдагчдын нэг Дамерау (Damerau) дэвшүүлсэн бөгөөд Левенштэйний алгоритмыг сайжруулж, өргөтгөөд Дамерау-Левенштэйн (Damerau-Levenshtein) гэж нэрлэсэн.

3.1.2 N-Gram

N-gram алгоритмын математик илэрхийлэл

Нийт магадлал $\hat{P}(w_i|w_{i-2}w_{i-1})$ нь 1 байх ёстой ба шинээр $\tilde{P}(w_i|w_{i-2}w_{i-1})$-г тодорхойлов. Иймд α_1, α_2 гэсэн 2 тогтмолыг ашиглан $\hat{P}(w_i|w_{i-2}w_{i-1})$ болон $\tilde{P}(w_i|w_{i-2}w_{i-1})$ хоёрын нийт магагдлал нь 1 болно.

$$\hat{P}(w_i|w_{i-2}w_{i-1}) = \begin{cases} \tilde{P}(w_i|w_{i-2}w_{i-1}), & \text{хэрэв } C(w_i|w_{i-2}w_{i-1}) > 0 \\ \alpha_1\tilde{P}(w_i|w_{i-1}), & \text{хэрэв } C(w_{i-2}w_{i-1}w_i) = 0 \text{ ба } C(w_{i-1}w_i) > 0 \\ \alpha_2\tilde{P}(w_i), & \text{бусад үед} \end{cases}$$

N-Gram арга нь хамгийн түгээмэл ашиглагдах аргуудын нэг юм. N-Gram-ыг ашиглах эхний алхам бол корпус ашиглан тухайн хэлэнд тохирох N-Gram-ыг олох явдал юм. Энэ аргыг шууд ашиглахын тулд үгнүүдийн бүх боломжит төрөл, холбоосыг агуулсан корпус ашиглан. Аль ч хэлний хувьд хязгаарлагдмал тооны давтамжтай n-gram байдаг бөгөөд олон төрлийн маш ховор үзэгддэг n-gramтай үгнүүд байдаг учраас бид бүхэн Back-off smoothing аргыг MED (Minimum Edit Distance) аргатай хамт ашигласан болно. Иймд, уг ховор n-gram-уудаас үүдэн Maximum Likelihood Estimation (MLE) аргыг ашиглан sparse matrix-ийг үүсгэнэ. Уг матриц дахь тэг утгуудыг дүүргэхийн тулд smoothing аргыг ашиглан. Үүнд, smoothing арга нь давтамжтай n-gram-уудыг цөөрүүлэх замаар давтагдаагүй n-gram-ийг гаргаж авахад ашиглагддаг ба нийт магадлал нь 1-тэй тэнцүү байх ёстой. Add-one Smoothing, Good-Turing Estimation, and Back-off smoothing гэх мэт өөр өөр төрлийн smoothing арууд байдаг ба эдгээр аргууд дотроос back-off smoothing арга нь хамгийн тохиромжтой байна.

Энэ аргыг ашиглан олон төрлийн алгоритм зохиож болохоор байна. Жишээ нь үгийн алдаа илрүүлэхэд хамгийн түгээмэл ашиглагддаг Байесийн арга нь тухайн үг бүр нь зөвхөн ганц алдаатай байна гэж үздэг. Гэхдээ доор өгөгдсөн MED алгоритм нь хоёр мөрийг харьцуулах ерөнхий арга юм. Энэ алгоритмыг мөр хоорондын зайг (ялгааг) тооцоолоход ашигладаг бөгөөд тухайн хоёр мөр хоорондоо хэр адилхан байгаа эсэхийг тогтоодог хэмжүүр юм.

Corpus (хөмрөг)

Mono-gram буюу нэг үгнээс, бүх хувирсан хэлбэрээр бүрдсэн, bi-gram буюу дараалсан 2 утгатай үгнээс бүрдсэн ба tri-gram буюу дараалсан 3-үгнээс бүрдсэн утгатай үгний сан юм. Уг гурван төрлийн санг давтамжаас нь хамааран back-off smoothing буюу n-gram-ын тоог багасгах замаар ашиглах юм.

Sparse Matrix (Тархалтын матриц)

Шинжлэх ухааны тооцон бодох салбарт ихэнхдээ шахалтын (compression) аргыг ашиглан тэгтэй, тэггүй бүх утгуудаа авахаас илүүтэй тухайн матрицын тэг биш утгуудыг авч жинхэнэ утгаа оновчтой аргаар гаргаж авна. Хэрэв компьютер технологи нь бусад бүх салбарт илүү үр дүнтэй ажиллаж байгаа бол (тооны машин, машин, галт тэрэг, анагаах ухаан г.м.) яагаад үүнийг текст боловсруулахад хэрэглэж болохгүй гэж? Үүнд дараах хоёр гол бэрхшээл байна.

1) Ямар ч хэлбэрээр их хэмжээний өгөгдөлд боловсруулалт болон дүн шинжилгээ хийх.
2) Компьютер өөрөө зөвхөн тоон утгаар харилцаж, ойлголцох боломжтой. Иймээс текстийг ойлгож, шинжилгээ хийхээс өмнө эхлээд уг текстийг компьютер ойлгож болох хэлбэр (тоо) рүү хөрвүүлсэн байх шаардлагатай.

Эхний асуудлын хувьд параллель тооцооллын арга болон GPU-ийн хүчэн чадлаар шийдэж болно. Харин хоёр дахь асуудлыг математикийн аргаар тархалтын матрицийг ашиглаж шийддэг. Математикт матриц гэдэг нь багана болон мөр хэлбэрээр өгөгдсөн бүх төрлийн өгөгдөл, мэдээллийг хэлдэг. Sparse буюу тархалттай матриц нь ихэнх мөр болон баганууд нь тэг, 0, утгатай матрицийг хэлдэг. Өөрөөр хэлбэл матрицын тэгтэй элементүүдийг тэг биш элементүүдэд хувааж тухайн матрицын тархалтыг утгыг олж болно.

$$\begin{bmatrix} 1.1 & 0 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 1.9 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 2.6 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 7.8 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.5 & 2.7 & 0 & 0 \\ 1.6 & 0 & 0 & 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.9 & 1.7 \end{bmatrix}$$

Зураг 21. Тархалтын матриц

Жишээ нь дараах хоёр өгүүлбэрийг авч үзье. Үүнд:

- Mary, is hungry for apples.
- John is happy he is not hungry for apples.

Нэгдүгээр алхам:

Цэг таслалуудыг арилгах

- Mary is hungry for apples
- John is happy he is not hungry for apples

Хоёрдугаар алхам:

Компьютер нь том ба жижиг үсэгнүүдийг ялгаж үздэг учраас өгүүлбэрт байгаа бүх үсэгнүүдээ жижиг үсэг болгоно.

- mary is hungry for apples
- john is happy he is not hungry for apples

Гуравдугаар алхам:

Өгүүлбэрт орсон бүх хосгүй үгнүүдээ зааглах (tokenize-токен зааглуур).

- [mary], [is], [hungry], [for], [apples]
- [john], [is], [happy], [he], [is], [not], [hungry], [for], [apples]

Дөрөвдүгээр алхам:

Бүх үгнүүдээ тоонд шилжүүлэх. Бүх үгнүүддээ тухайн өгүүлбэрт давтсан давтамжийн тоог оноодог.

Зураг 22. Тархалтын матрицын давтамж

Иймд, уг ховор n-gram-уудаас үүдэн Maximum Likelihood Estimation (MLE) аргыг ашиглан тархалтын матрицыг үүсгэнэ. Уг матриц дахь тэг утгуудыг дүүргэхийн тулд дараах smoothing аргыг ашиглан. Энэ аргыг үгийн алдаа шалгах программд ашиглахдаа бүх n-gram-ын баазыг бий болгож, үүнд давтамжийн тоог индексжүүлж ашиглана.

Smoothing аргууд

Smoothing арга нь давтамжтай n-gram-уудыг цөөрүүлэх замаар давтагдаагүй n-gram-ыг гаргаж авахад ашиглагддаг ба нийт магадлал нь 1-тэй тэнцүү байх ёстой. Add-one Smoothing, Good-Turing Estimation, and Back-off smoothing гэх мэт өөр өөр төрлийн smoothing арууд байдаг ба эдгээр аргууд дотроос back-off smoothing арга нь хамгийн тохиромжтой байна.

Katz (Back-Off) smoothing

Энэ арга нь хэрэв n-gram нь 0-давтамжтай байх юм бол (n-1)-gram-ыг ашиглах юм. Жишээ нь: Доорх томъё-1-д харуулснаар хэрэв тухайн trigram нь олдоогүй тохиолдолд, bigram-ыг, bigram нь олдоогүй бол monogram-ыг ашиглан sparse matrix-г дүүргэх юм. Иймд n-gram загварыг monogram болтол нь (n-1), (n-2) г.м. байдлаар бууруулж зохион байгуулна.

Энэ аргыг ашиглан олон төрлийн алгоритм зохиож болохоор байна. Жишээ нь үгийн алдаа илрүүлэхэд хамгийн түгээмэл ашиглагддаг Байесийн арга нь тухайн үг бүр нь зөвхөн ганц алдаатай байна гэж үздэг. Гэхдээ доор өгөгдсөн MED алгоритм нь хоёр мөрийг харьцуулах ерөнхий арга юм. Энэ алгоритмын тухай дээр дэлгэрэнгүй дурдсан болно.

3.1.3 Машин сургалтын алгоритм

Машин сургалтын алгоритмууд урьдчилан тодорхойлогдсон томъёо, загваргүйгээр, зөвхөн тооцоолох аргад суурилан, өгөгдлөөс шууд “суралцаж” мэдээллийг олж авдаг. Суралцах өгөгдөл ихсэхийн хэрээр алгоритмын ажиллагаа улам сайжирдаг. Гэвч машин сургах арга нь дан ганц өгөгдлийн сангийн асуудал биш бөгөөд хиймэл оюун ухааны нэгэн хэсэг юм. Систем ухаантай байснаар орчноо солиход тухайн байдалдаа суралцах боломжтой байдаг [8]. Хэрэв систем суралцаж чаддаг, өөрчлөлт үзүүлж чаддаг л бол хөгжүүлэгчийн зүгээс боломжит бүх нөхцөлүүдэд гаргах шийдлүүдийг оруулж өгөх шаардлагагүй болно.

1. Supervised Learning буюу Удирдлагатай сургалт

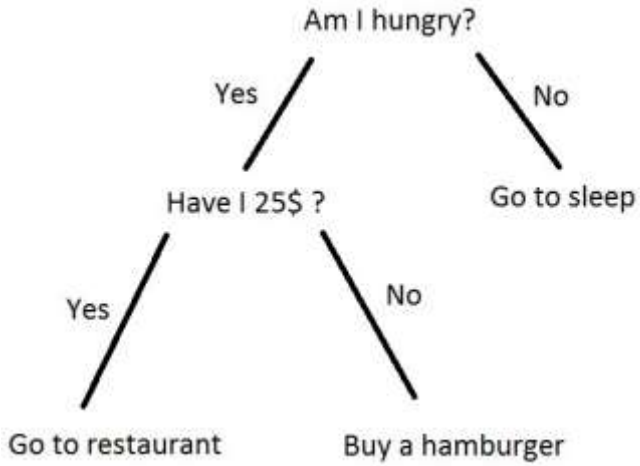
Хэв шинжийг олохдоо оролтын болон гаралтын өгөгдлийг хоёуланг нь ашигладаг аргыг хэлдэг. Бүх удирдлагатай сургалтын аргууд Classification (Ангилал) болон Regression (Регресс) техникийн аль нэгт хамаарна. Ангиллыг салангид (discrete) утга таамаглахад ашигладаг. Жишээ нь: дэмжсэн баг маань ЯЛАХ уу ЯЛАГДАХ уу? Энэ имэйл СПАМ уу, ЖИНХЭНЭ үү? гэх мэт асуултын хариу оролтоосоо хамаарч ЯЛАХ, ЯЛАГДАХ эсвэл СПАМ, ЖИНХЭНЭ гэсэн салангид бүлгүүдэд хамаарна. Регрессийг үргэлжилсэн (continuous) утга таамаглахад ашигладаг. Жишээ нь, хувьцааны зах зээлийн үнийн хэлбэлзэл, цаашдын хандлага; цаг агаарын урьдчилсан таамаг гэх мэт.

2. Unsupervised Learning буюу Удирдлагагүй сургалт

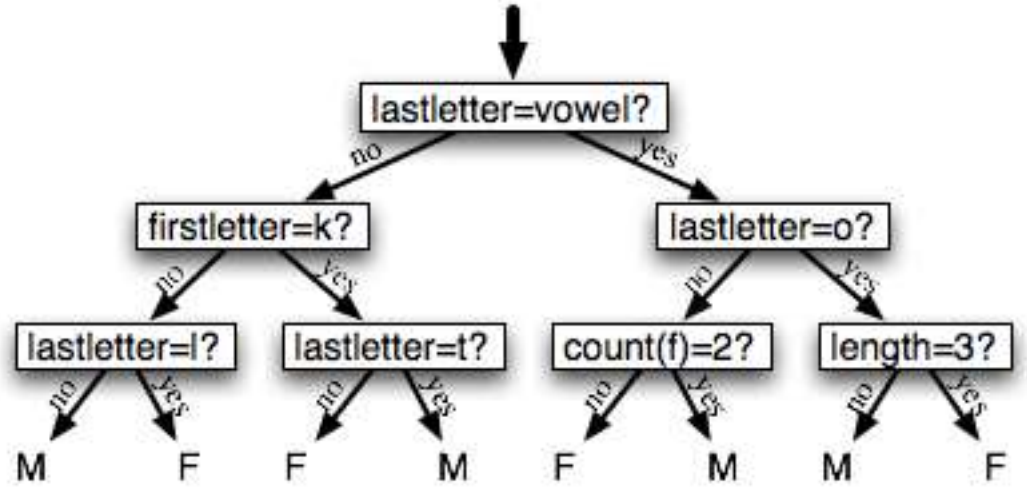
Хэв шинжийг олохдооз өвхөн оролтын өгөгдлийг ашигладаг аргыг хэлдэг. Өгөгдлөөс яг юу хайхаа мэдэхгүй байгаа үед энэ арга тохиромжтой. Түүхий өгөгдлийг ойлгохын тулд ихэвчлэн ашигладаг. Ихэнх удирдлагагүй сургалт Cluster Analysis (Бүлэглэх шинжилгээ) гэдэг техник дээр суурилна. Бүлэглэх шинжилгээ гэдэг нь, өгөгдлийн шинж чанаруудыг хэмжээд, адил төстэйгээр нь бүлэглэж хуваахыг хэлнэ.
Машин сургах аргуудаас дурдвал. *Үүнд:*

Шийдвэрийн мод

Шийдэврийн мод бол боолен функцын нэг дүрслэл юм. Тухайн 1 хугацааны шийдвэрийн жагсаалт нь дизъюнкц /тусгаарлах/ болон нэгдэл байдлыг илүү илэрхийлдэг. Гэсэн хэдий ч, 1 хугацааны шийдвэр жагсаалт ерөнхий тусгаарлах хэвийн хэлбэр, хосолмол хэвийн хэлбэрээс бага байдлыг илэрхийлдэг байна. к урттай шийдвэр жагсаалтын хувьд заасан хэл нь дэд олонлог гэж нэрлэгдэх к-гүн буюу шийдвэр модны заасан хэлийг агуулна [16][17]. Шийдвэрийн мод нь машин сургах аргад түгээмэл ашиглагддаг бөгөөд илүү сайн үр дүн өгдөг аргуудын нэг юм.



Зураг 23. Шийдвэрийн модны жишээ



Зураг 24. Шийдвэрийн модыг хэл шинжлэлд ашиглах нь

Нейроны сүлжээ

Биологийн үндэслэл дээр үндэслэн хиймэл нейроны сүлжээний санааг 1943 онд МакКуллош Пит нар нээжээ. 1986 онд Румэлхарт, МакКелланд нар Англи хэлний өнгөрсөн цагийн хувиллын хэлбэрүүдийг нейроны сүлжээгээр туршиж үзжээ. Түүнээс хойш үгийн хувиллыг нейроны сүлжээгээр гүйцэтгэх, үгийн бүтцийг нейроны сүлжээгээр тодорхойлох оролдлого олон хийгдэж байсан. Компьютер олон үйлдлийг хүнээс харьцануй хурдан хийдэг ч компьютерын шийдэж чадахгүй олон асуудлыг балчир хүүхэд ч хийж чадна. Жишээлбэл, хонь ямаа хоёрыг хүүхэд ялгаж чадна. Хүний тархины биологийн нейроны сүлжээг дуурайлган хиймэл нейроны сүлжээг компьютерт дүрслэхийг Хиймэл нейроны сүлжээ гэж нэрлэдэг.

Энгийн алхмуудтай, хийх гэж байгаа үйлдлүүд нь тодорхой, төлөв нь тогтвортой алгоритмтай программд нейроны сүлжээ ашиглах шаардлагагүй. Тодорхой тооны алхмуудаар илэрхийлэх боломжгүй, тогтворгүй төлөвтэй асуудлыг шийдвэрлэхэд нейроны сүлжээ тохиромжтой. Үүнд, дүрс таних, ангилах, өгөгдлийн агуулах гэх мэтийн

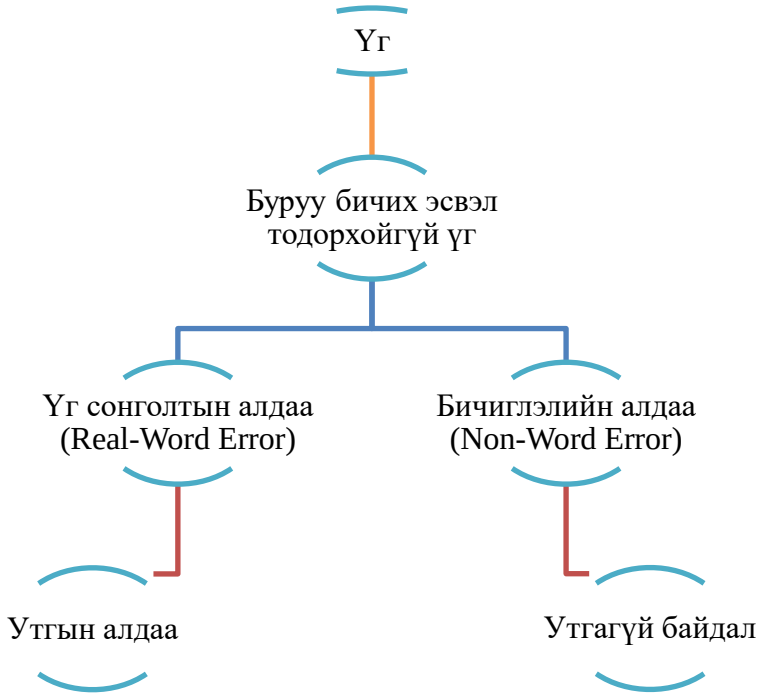
асуудлууд орно. Нейроны сүлжээг ашигласнаар уламжлалт программчлалын алгоритмыг бодвол программын багахан хэмжээний кодчилол шаардагддаг [18].

Нейроны сүлжээг сургах хоёр үндсэн арга байдаг. Үүнд:

- Удирдлагатай сургах арга
- Удирдлагагүй сургах арга

Удирдлагатай сургах арга нь урьдчилан тодорхойлсон гаралтууд бүхий жишээ өгөгдлийн олонлогоор нейроны сүлжээг сургадаг. Энэ арга нь хамгийн түгээмэл хэлбэр юм.

3.2 Үгийн алдааг олох арга, алдааны төрөл



Зураг 25. Үгийн алдааны төрөл

- Доод түвшний ажил (Алдаа шалгагч):
  - Буруу үгийг олох (NWE);
  - Зөв үгийг санал болгож, эрэмбэлэх
  - Автомат эсвэл гараар засах
- Дээд түвшний ажил (Үг сонголтын алдаа-RWE засах):
  - Үг зөв мөртлөө утгын алдаатай үгийг олох (RWE)
  - Зөв үгийг санал болгож, эрэмбэлэх
  - Автомат эсвэл гараар засах

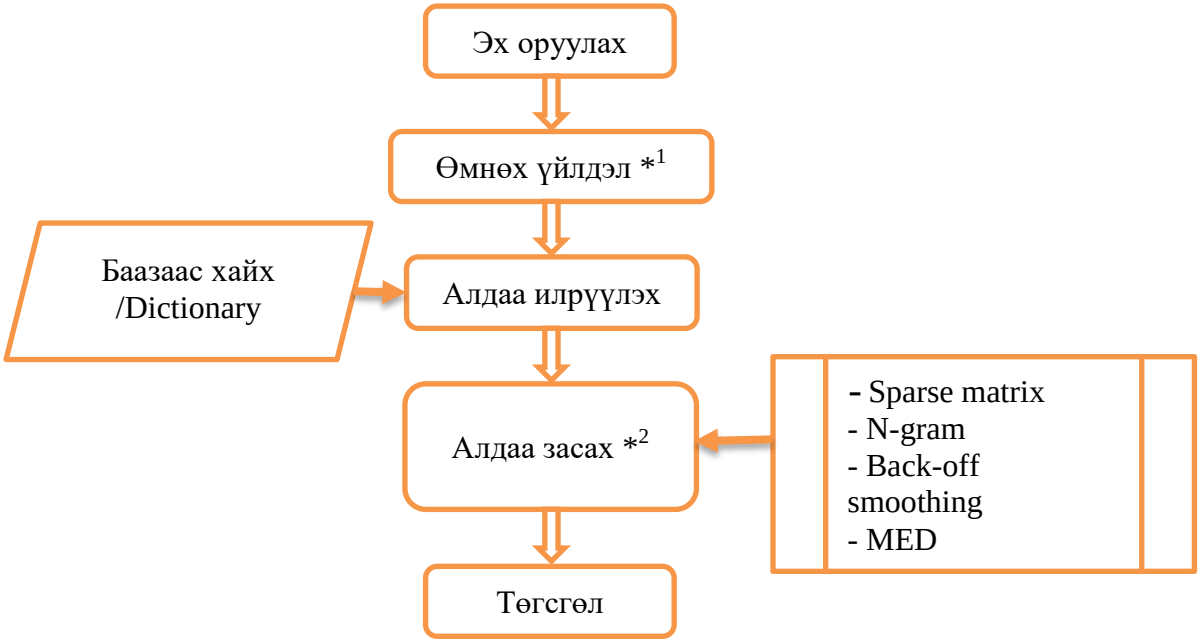
Үг биш алдааны төрлүүд (NWE):

1. Салгаж бичигдэх (Split word): Үгийн дотор алдаатайгаар хоосон зай орхих.
2. Нийлж бичигдэх (Run-on or Merged words): Хоёр эсвэл түүнээс дээш үгийг нийлүүлж бичих.

3. Үсэг нэмэх, хасах, эсвэл орлуулах (insertion, deletion and substitution - IDS): Нэг ба түүнээс дээш үсэг нэмж, хасаж, эсвэл орлуулж бичих.

Жич: Алдааг илрүүлэх арга бол үгийн жагсаалтаас хайх буюу Dictionary lookup аргыг ашиглана. Хэрэв адилхан үг олдохгүй бол буруу үг ба алдаа засах алхам руу орно.

Үгийн алдаа засах



Зураг 26. Үгийн алдаа засах алхамууд

***1- Өмнөх үйлдэл:** Өмнөх үйлдэл нь алдаа шалгахаас өмнөх алхам юм. Энэ алхамд тухайн оруулсан үгийг товчилсон үг, том жижиг үсгийн ялгаа болон бусад тэмдэгтүүдийг ялгаж салгах юм.

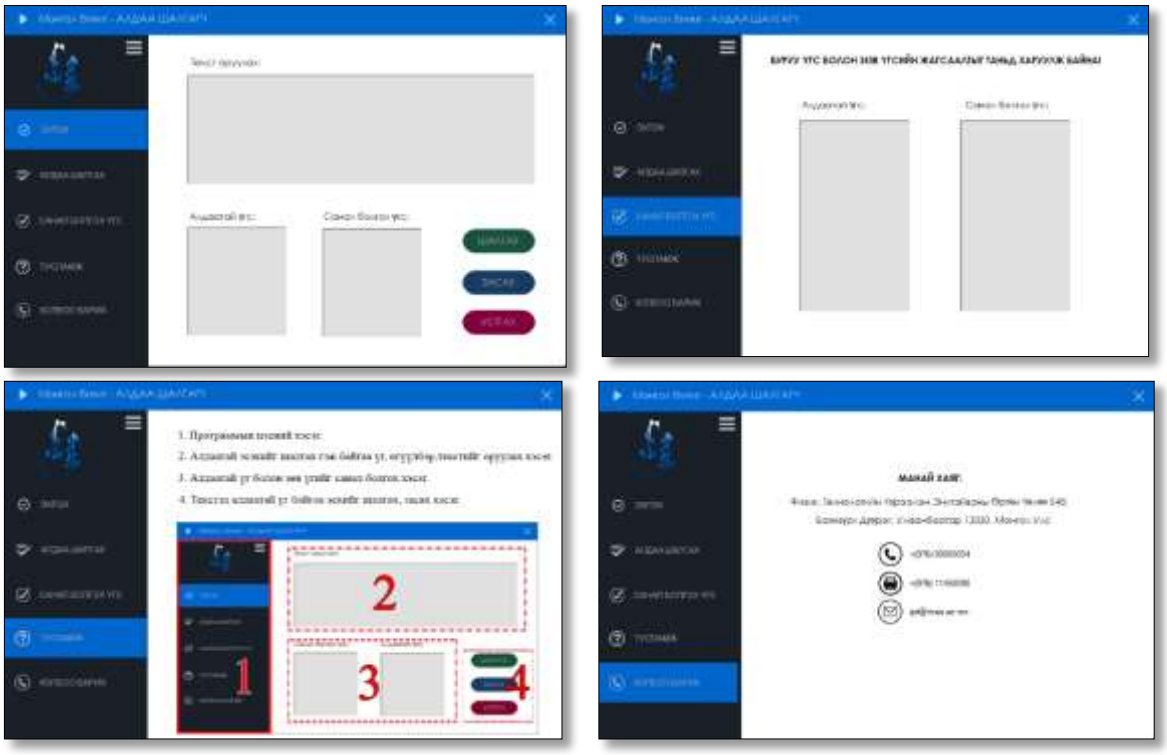
- Өгүүлбэрийн төгсгөлийг ялгах. Хэрэв тухайн үг нь цэг (.)-ээр дууссан бол өгүүлбэрийн эцсийн үг эсэхийг шалгаж тухайн үгнээс цэгийг салгаж шалгах. Энэ арга нь мөн адил бусад тэмдэгтүүдэд бас үйлчилнэ.
- Товчилсон үгийг таних. Товчилсон үгийг том үсгээр бичигдсэн эсвэл үсэг хооронд цэг бичигдсэн байдлаар танина.

***2- Үгийн алдааг засах:** N-Gram арга нь үгийн алдааг засах алхамд ашиглагдах юм. Гэхдээ энэ арга нь дангаар биш Katz-ын back-off smoothing арга болон MED (Minimum Edit Distance) гэсэн аргуудын хослуулан ашиглах юм. Үгийн алдаа засах аргууд болон алхмуудын талаар дараагийн бүлэгт дэлгэрэнгүй оруулав.

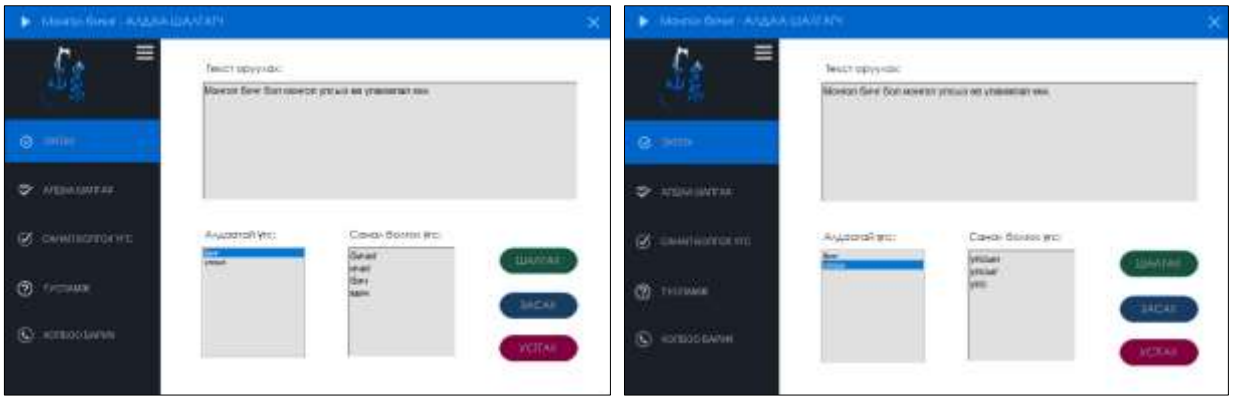
Алдаа засах алхам:

N-Gram аргыг ашиглах: үгийн болон утгын алдааг N-Gram аргыг backoff smoothing ба Minimum Edit Distance (MED) аргуудтай цуг ашиглана. Үүнд, эхлээд 3-Gram хэрэв олдохгүй бол дараа нь 2-Gram ба олдохгүй бол дараа нь Mono-Gram гэсэн smoothing арга ашиглана.

3.3 Тушилт, үр дүн



Зураг 26. Алдаатай үг олох программ хангамжийн интерфэйс



А. Левенштэйний алгоритмын үр дүн.

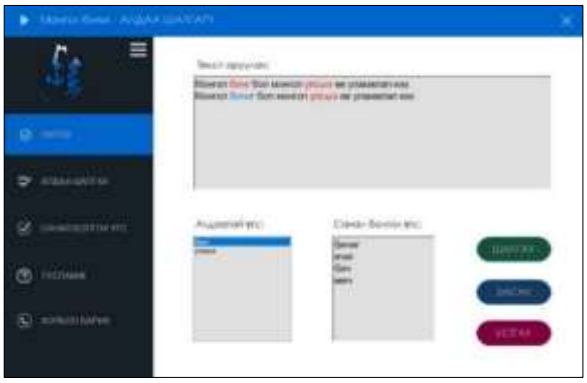
Зураг 27. Левенштейны алгоритмын үр дүн

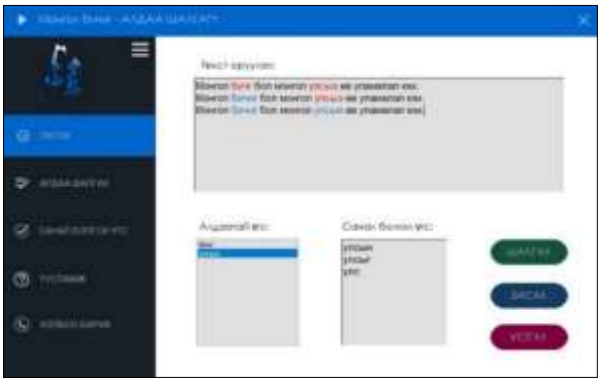
Бичвэрийн алдааг шалгахадаа **“шалгах”** гэсэн товчыг дарахад алдаатай үгнүүдийн жагсаалт гарч ирнэ. Тухайн алдаатай үгнүүд дээр дарахад зөв байж болох үгийн жагсаалт гарч ирнэ.

Жишээ нь: “Монгол бичг бол монгол улсыэ өв уламжлал юм.”

Энэ өгүүлбэрээс 2 үг алдаатай байгаа тэр 2 үгийг **алдаатай үгс** гэсэн хэсэгт гаргаж аваад тухайн үгэн дээр дарахад хажуу талын **санал болгох үгс** гэсэн хэсэгт зөв байж болох үгнүүдийг жагсаалтыг харуулсан байгаа. Жишээлбэл:

“бичг” гэдэг үг дээр гэхэд “бичиг”, “ичиг”, “бич”, “мич” гэсэн энэ үгнүүд байж болох юмаа гэж санал болгож байна.

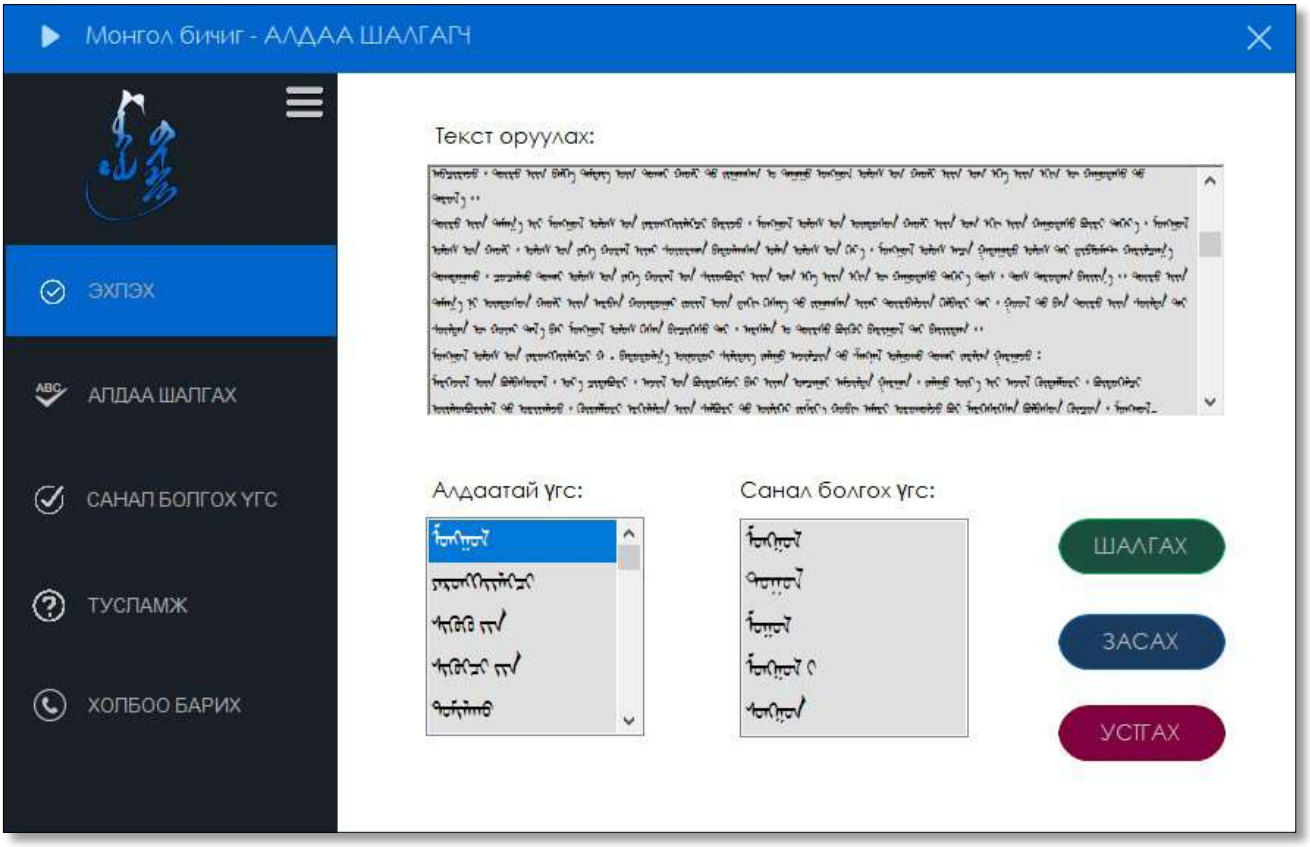




“улсыэ” гэдэг үг дээр гэхэд “улсын”, “улсыг”, “улс” үгнүүдийг санал болгож байна.

Зураг 28. Буруу үгийг засах

Туршилт-1.1



Зураг 29. Программын хэсэг ба Туршилт 1.1

Туршилтын үр дүнг news.mn сайтын 50 мэдээ, president.mn сайтын 20 мэдээ, http://talchir.com 3 нийтлэл, http://mongol-bichig.dusal.net/-ээс 12 ш үлгэр нийт 85 нийтлэлийг авч шалгасан.

Хүснэгт 14. Туршилт 1.1-ийн үр дүн

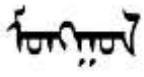
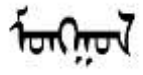
Нийт нийтлэлийн тоо	Нийт үгийн тоо	Алдаатай үг		Санал болгосон үг
		Үгийн санд байхгүй үг	Алдаатай үг	
85	11,759	293	101	1970

		394	

Алдаатай үг болгон дээр хамгийн ойролцоо зөв байж болох 5 үгийг санал болгож байгаа бөгөөд 394 алдаа үг дээр 1970 үгийг санал болгож байгаа юм. Үгийн санд байхгүй бол зөв бичигдсэн үгийг алдаатай гэж үзнэ. Тэгээд тэр үгтэй хамгийн ойролцоо үгийг санал болгон. Үүнээс юуг хэлж болохов гэхээр программын үр дүн санд байгаа үгнээс шууд хамааралтай учир хувьлах нь дутагдалтай тийм болхоор үр дүнгээ хувьлаагүй болно.

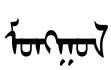
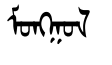
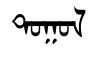
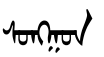
Жишээ нь №1.1. Юникодын хувьд аваад үзэхэд 1829 (н) гэсэн код гарах ёстой ч эхний алдаатай гэж үзсэн монгол гэдэг үг дээр юникодыг танихгүй байсан. Тийм учраас программ нь буруу гэж үзээд сангаас зөв бичигдсэн монгол үгийг санал болгосон.

Хүснэгт 15. Үгийн алдааны харьцуулалт

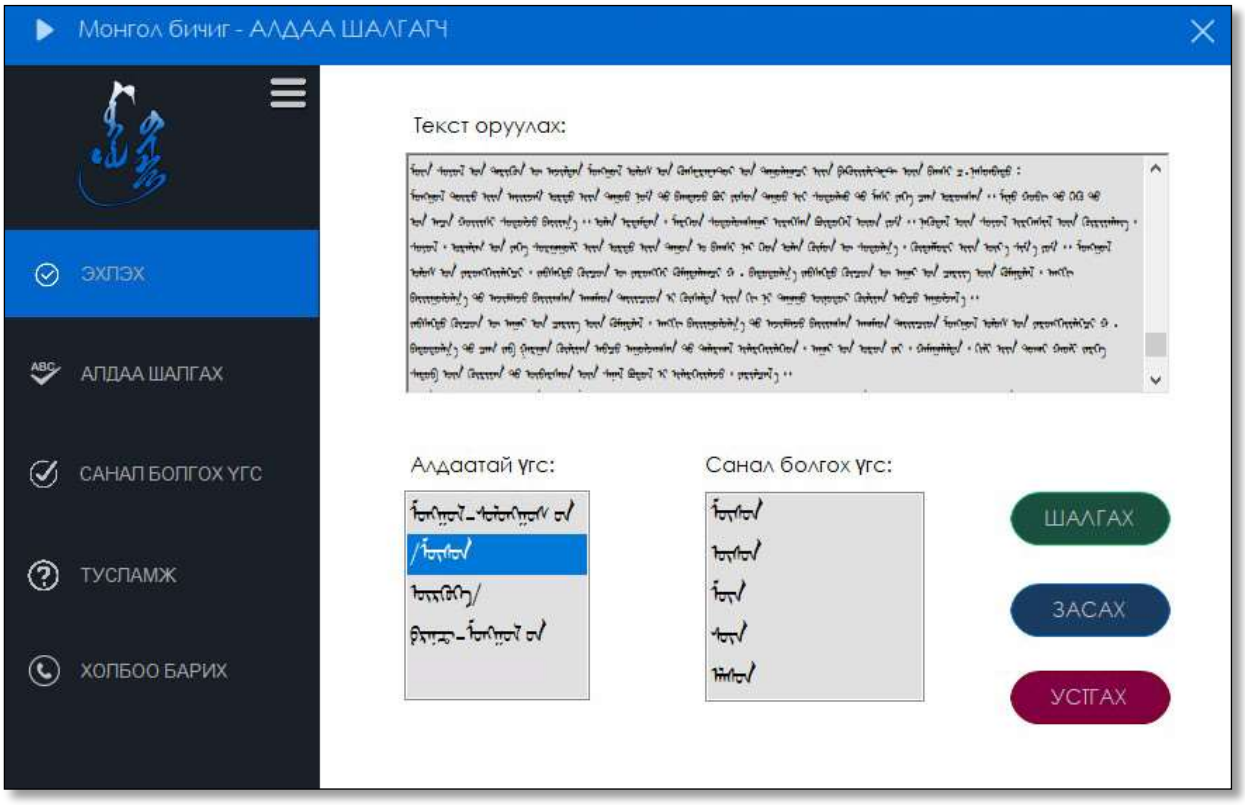
Үг	**Юникод**	**Төрөл**
	182E1823 ᠍ 182D1823182F	Алдаатай үг
	182E1823**1829**182D1823182F	Санал болгосон зөв үг

Яагаад дээд тал нь буруу байна вэ гэвэл эр үгийн Г-ээ бичихдээ shift-тэй даралгүй бичсэн байна. Учир нь Shift даралгүй бичвэл илүү нуруу гардаг.

Хүснэгт 16. Туршилт 1.1-ийн үр дүнгийн хэсгээс

Буруу үг	**Санал болгосон үгс**
	    

Туршилт-1.2



Зураг 30. Программын хэсэг ба Туршилт 1.2

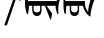
101 алдаатай үгийг засан, дээрх нийтлэлийг санд нэмж дахин шалгаж үзэхэд:

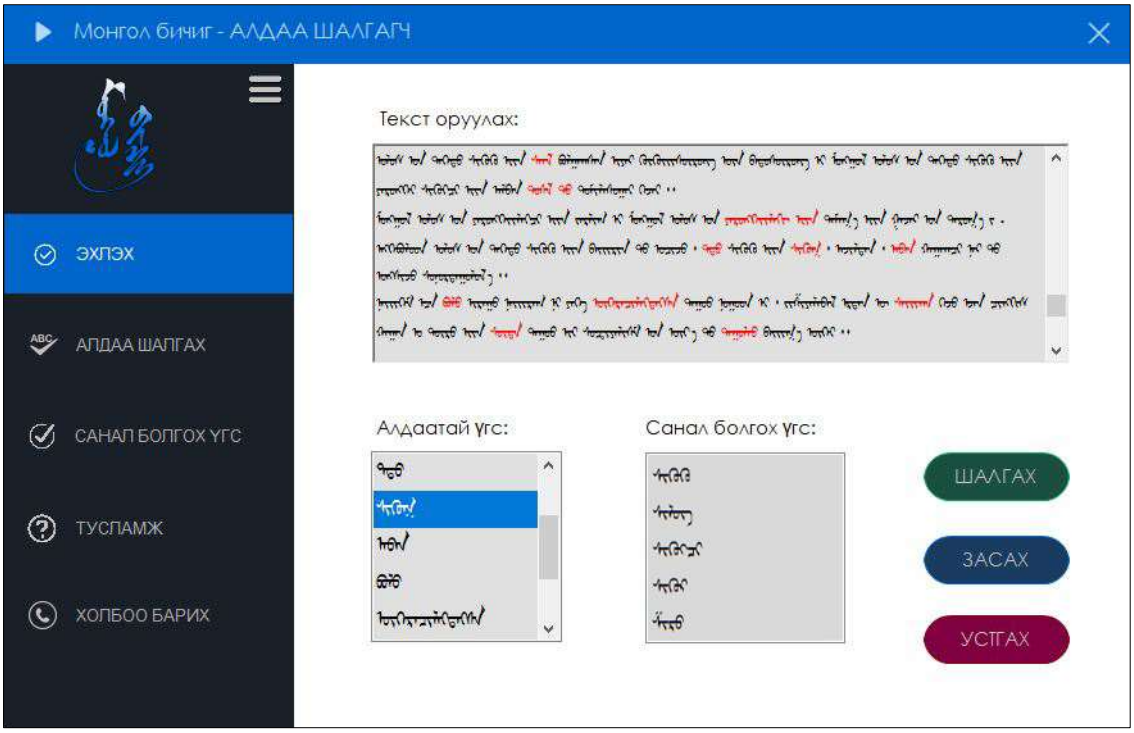
Хүснэгт 17. Туршилт 1.2-ын үр дүн

Нийт нийтлэлийн тоо	Нийт үгийн тоо	Алдаатай үг		Санал болгосон үг
		Үгийн санд байхгүй үг	Алдаатай үг	
85	11,759	0	4	20
		4		

4 үгийг бүрэн засаж чадаагүй байна (*зураг 30*). (мөсөн) гэдэг үг урд \ тэмдэгттэй, (өргөө) гэдэг үг ардаа / тэмдэгтэй, , гэсэн хоёр үг нь - тэмдэгтэй 4 үгийг алдаатай гэж үзсэн байна.

Хүснэгт 18: Туршилт 1.2-ын үр дүнгийн хэсгээс

Буруу үг	**Санал болгосон үгс**
/ 	   



Туршилт-1.3

Зураг 31. Программын хэсэг ба Туршилт 1.3

11 үгийг алдаатай бичиж шалгахад:

Хүснэгт 19. Туршилт 1.3-ын үр дүн

Нийт нийтлэлийн тоо	Нийт үгийн тоо	Алдаатай үг		Санал болгосон үг
		Үгийн санд байхгүй үг	Алдаатай үг	
85	11,759	0	11	55
		11		

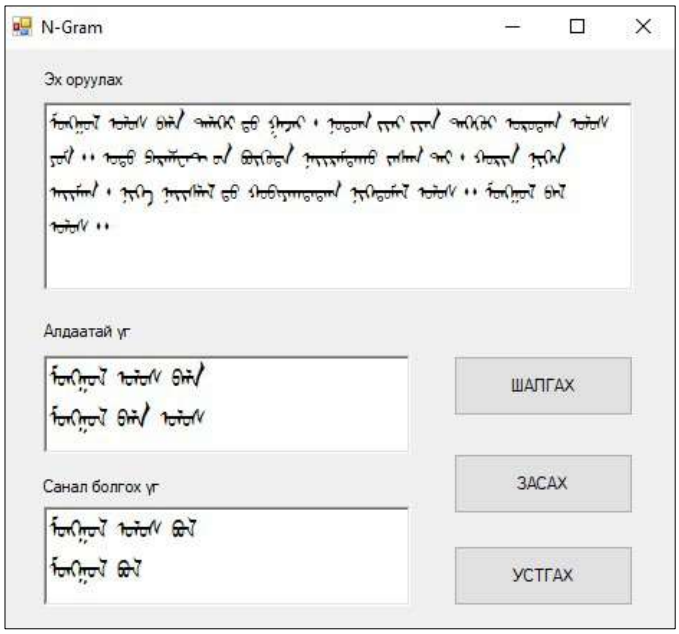
Хүснэгт 20. Туршилт 1.3-ын үр дүнгийн хэсгээс

Буруу үг	**Санал болгосон үгс**

Б. N-GRAM -ийн үр дүн**.**

- Үг зөв боловч утгын алдаатай үгийг олох, засах

Туршилт-2.1



Зураг 32. Программын хэсэг ба Туршилт 2.1

N-Gram аргыг ашиглах давуу тал нь үгийн утгын алдааг илрүүлж олох боломж юм. Мөн санд суурилсан учраас сангаа өргөжүүлэх тусам илүү сайн сурна. N-Gram арга нь эхийг өгүүлбэр өгүүлбэрээр шалгадаг.

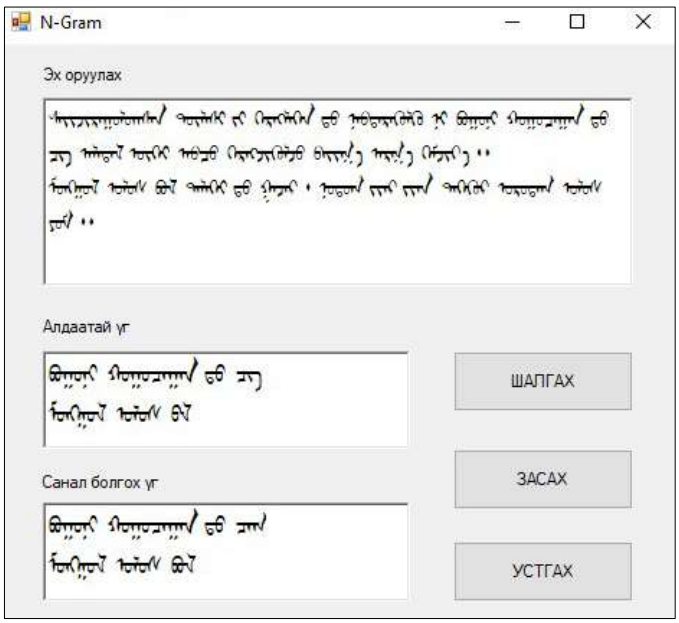
Алдаа-1

“Монгол улс **бал** дэлхийд газар, нутгаараа дээгүүр ордог улс юм” гэсэн эхний өгүүлбэр дээр “монгол улс бал” гэсэн trigram бодоод “монгол улс **бол**” гэдэг хэлцийг санал болгож байна.

Алдаа-2

“Монгол **бал** улс” гэдэг өгүүлбэр дээр trigram бодоод үр дүн олдохгүй бол bigram бодно. “Монгол **бал**” гэдэг хоёр үг дээр bigram бодоод гарсан үр дүнг санал болгоно. Монгол бал гэж байж болох ч хамгийн их хэрэглэдэг буюу давтамжийн тоо нь хамгийн өндөр bigram-ийг санал болгоно. Бидний жишээн дээр “Монгол **бол**” гэж санал болгосон.

Туршилт-2.2



Зураг 33. Программын хэсэг ба Туршилт 2.2

Оруулсан эхэд trigram бодоод дараах 2 үр дүнг алдаатай гэж илрүүлж, зөв байж болох үр дүнг санал болгосон.

“Богино хугцаанд **цг**” - “Богино хугцаанд **цаг**”

“Монгол улс **бл**” - “Монгол улс **бол**”

Хүснэгт 21. Туршилт 2.1,2-ын үр дүн

Туршилт	Өгүүлбэрийн тоо	Нийт үгийн тоо	Алдаатай		Санал болгосон үг	
			bigram	*trigram*	*bigram*	*trigram*
Туршилт-2.1	15	340	1	1	1	1
Туршилт-2.2	13	264	0	2	0	2

Хүснэгт 22. Үр дүнгийн харьцуулалт

№	Алаадтай өгөгдөл	МТШХХГ корпус	Манай корпус		ӨМИСургууль корпус
		spellcheck.gov.mn	**Levenshtein**	**N-Gram**	http://mc.mglip.com:8080/
	(Монгол улс бол)	*Алдаагүй*	*Алдаагүй*	*Алдаагүй*	
1	(Монгол улс бл)	бөл бүл бол ба			
2	(Монгол улс бал)	*Алдаагүй*	*Алдаагүй*		
3	(Монгол бал улс)	*Алдаагүй*	*Алдаагүй*		
4	(Богино хугацаанд цг)	Гц Цаг Цог Цуг цэг			

Бичвэрийн үг, үсгийн зөв бичиглэлийг шалгах, алдааг засах процессыг хийхийн өмнө эхийг өгүүлбэрээр задлах, үг бүрээр задлах, хэрэггүй тэмдэгтийг гээх зэрэг урьдчилсан боловсруулалт хийх шаардлагатай байдаг. Эдгээр урьдчилсан боловсруулалтыг хийхийн тулд дараах программ хангамжуудыг боловсрууллаа.

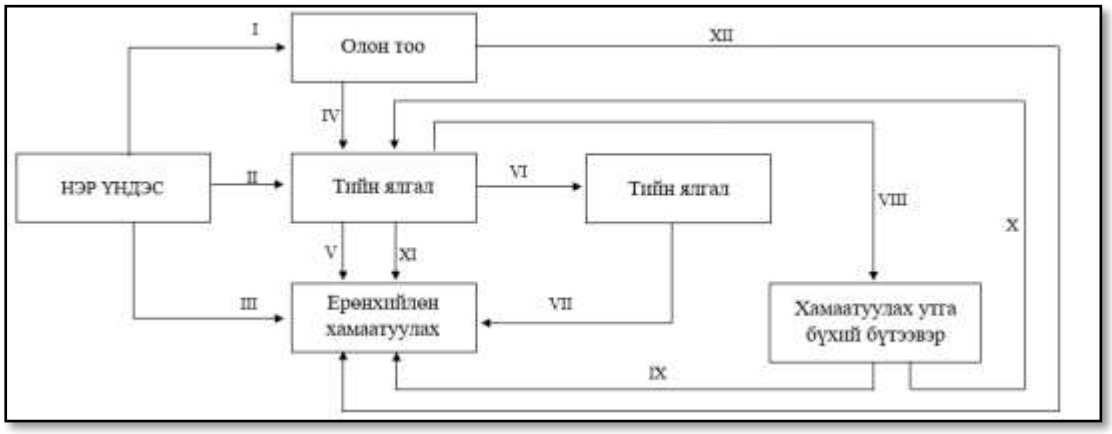
Холбогдох программ хангамжийн хөгжүүлэлт

Зорилго: Доорх 3 программ хангамжийг төслийн гол ажил болох **үг зүйн загварчлал, хоёр бичгийн алдаа шалгуур, хоёр бичгийн хөрвүүлэг** гэсэн программ хангамжуудад хэрэглэгдэх өгөгдлийн санг бүрдүүлэхэд ашиглах зорилгоор хөгжүүлэлт хийж ашиглав.

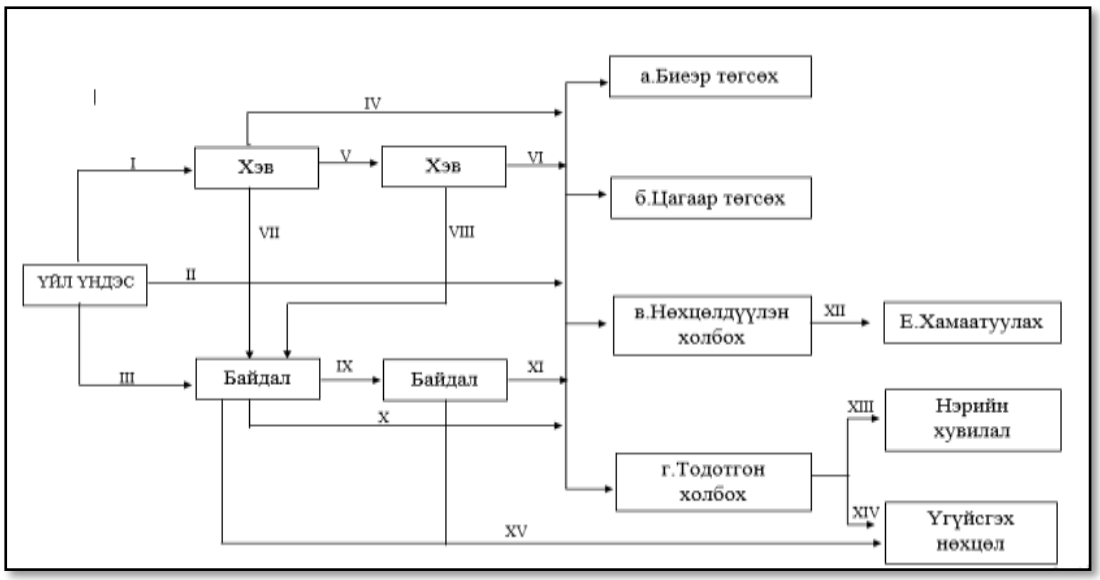
Нөхцөлийн сан бүрдүүлэх программ хангамж

Монгол хэлний нэр, үйл үгийн нөхцөлүүд, тэдгээрийн боломжит хослолоор өгөгдлийн сан үүсгэж, холбогдох программ хангамжийг хөгжүүлэв.

- 2 давхар нөхцөл – 5
- 3 давхар нөхцөл – 6
- 4 давхар нөхцөл – 4
- 5 давхар нөхцөл – 1



Зураг 34. Нэр үндсийн хувилах загвар



Зураг 35. Үйл үндсийн хувилах загвар

Нэр үндэсийн хувьд нийт давхарлан орох боломжтой нөхцөл 16 ширхэг байна гэдгийг дээрх хүснэгтээс тооцоолж гаргаж ирсэн.

- 2 давхар нөхцөл -16
- 3 давхар нөхцөл - 27
- 4 давхар нөхцөл - 32
- 5 давхар нөхцөл -26
- 6 давхар нөхцөл - 15
- 7 давхар нөхцөл - 8
- 8 давхар нөхцөл - 4
- 9 давхар нөхцөл - 1

Үйл үндэсийн хувьд нийт давхарлан орох боломжтой 129 нөхцөл байна.

Нэг давхар нөхцөлийн санг ашиглаж бусад 2, 3, 4, ..., 9 давхар нөхцөлүүдийн боломжит хослолын санг гаргаж авсан. Нэг давхар сангийн бүтцийг доорх жишээгээр үзүүлэв.

Тийн ялгал		
NC411	аас	
NC412	оос	
NC413	өөс	
NC414	ээс	
NC511	аар	
NC513	оор	
NC513	өөр	
NC514	ээр	
NC611	тай	
NC612	той	
NC613	тэй	
NC614	тэй	

Ерөнхий хамаатуулах		
NX111	аа	
NX112	оо	
NX113	өө	
NX114	ээ	
NX201	минь	
NX202	маань	
NX211	чинь	
NX212	тань	
NX221	нь	

Хэвийн нөхцөл		
VE121	лга	
VE122	лго	
VE123	лгө	
VE124	лгэ	
VE131	га	
VE132	го	
VE133	гө	
VE134	гэ	
VE141	аа	
VE142	оо	
VE143	өө	
VE144	ээ	

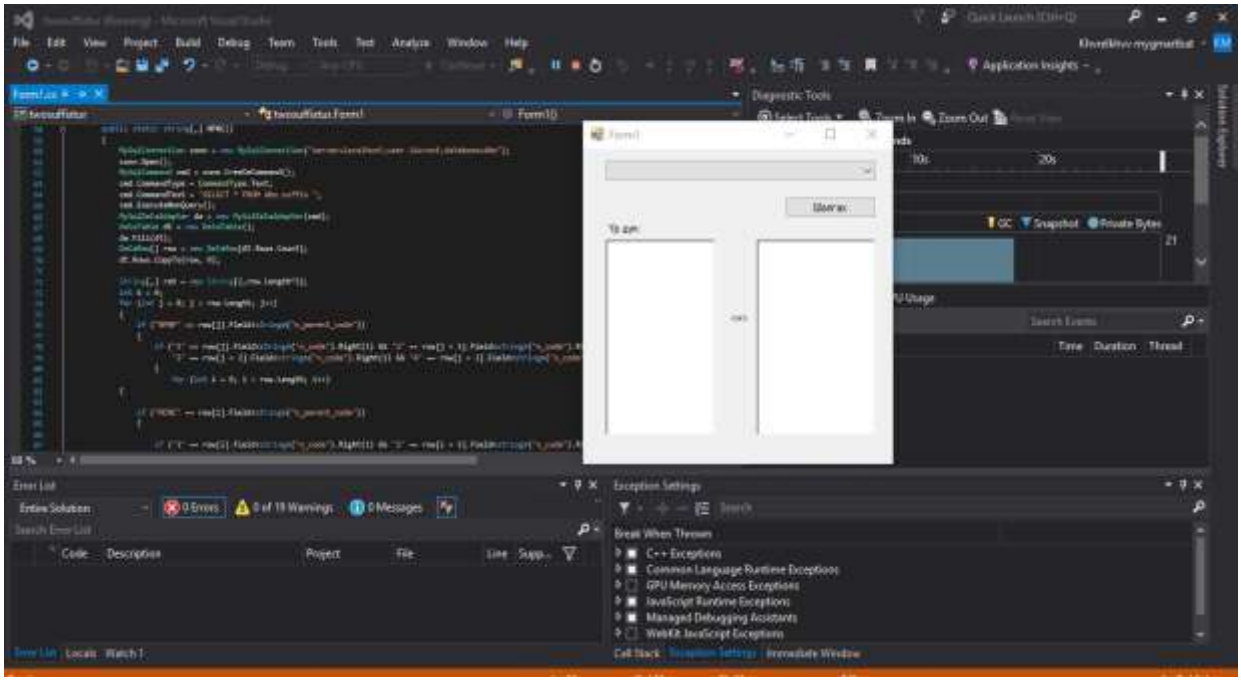
Байдлын нөхцөл		
VI311	аадах	
VI312	оодох	
VI313	өөдөх	
VI314	ээдэх	
VI321	зна	
VI322	зно	
VI323	знө	
VI324	знэ	
VI411	цгаа	
VI412	цгоо	
VI413	цгөө	
VI414	цгээ	

Зураг 36. Нэр, үйл үгийн нэг дан нөхцөл

Үндсэн өгөгдлийн сан нь нийт тоо давхардсан байдлаар нийт 1461899 давхарлан орох боломжтой нөхцөл байгаагаас нэр үг нөхцөл давхардсан тоогоор 35932, үйл үг нөхцөл давхардсан тоогоор 1425967 байна.

Хүснэгт 23. Программын үр дүн

Нэр үг (нөхцөл давхарлан орсон тоо)	Үйл үг (нөхцөл давхарлан орсон тоо)
▪ 2 давхар - 1244ш ▪ 3 давхар - 5808ш ▪ 4 давхар - 11600ш ▪ 5 давхар - 17280ш	▪ 2 давхар - 4224 ▪ 3 давхар - 30716 ▪ 4 давхар - 139465 ▪ 5 давхар - 396684 ▪ 6 давхар -282176 ▪ 7 давхар - 265782 ▪ 8 давхар - 161480 ▪ 9 давхар - 145440
Нийт 35932 ширхэг боломжит хосолсон үр дүн гарсан.	Нийт 1425967 ширхэг боломжит хосолсон үр дүн гарсан.



Зураг 37. Программын код болон ажиллаж байгаа хэсэг

```
DataTable dt = new DataTable();
da.Fill(dt);
DataRow[] row = new DataRow[dt.Rows.Count];
dt.Rows.CopyTo(row, 0);

String[,] ret = new String[2,row.Length*4];
int k = 0;
for (int i = 0; i < row.Length; i++)
{

    if ("NCNC" == row[i].Field<string>("n_parent_code"))
    {
        if ("1" == row[i].Field<string>("n_code").Right(1) && "2" == row[i + 1].Field<string>("n_code").Right(1) &&
            "3" == row[i + 2].Field<string>("n_code").Right(1) && "4" == row[i + 3].Field<string>("n_code").Right(1))
        {
            for (int j = 0; j < row.Length; j++)
            {
                if ("NXNX" == row[j].Field<string>("n_parent_code"))
                {
                    if ("11" == row[j].Field<string>("n_code").Right(2) && "12" == row[j + 1].Field<string>("n_code").Right(2) &&
                        "13" == row[j + 2].Field<string>("n_code").Right(2) && "14" == row[j + 3].Field<string>("n_code").Right(2))
                    {
                       if ("н" == row[i].Field<string>("n_seq_cyr").Right(1) || "ы" == row[i].Field<string>("n_seq_cyr") ||
                            "ий" == row[i].Field<string>("n_seq_cyr") )
                        {
                            ret[1,k] += row[i].Field<string>("n_seq_cyr") + "x" + row[j].Field<string>("n_seq_cyr");
                            ret[1,k + 1] += row[i + 1].Field<string>("n_seq_cyr") + "x" + row[j + 1].Field<string>("n_seq_cyr");
                            ret[1,k + 2] += row[i + 2].Field<string>("n_seq_cyr") + "x" + row[j + 2].Field<string>("n_seq_cyr") ;
                            ret[1,k + 3] += row[i + 3].Field<string>("n_seq_cyr") + "x" + row[j + 3].Field<string>("n_seq_cyr") ;
```

Зураг 38. Тийн ялгал + Е.хамаатуулах залгаж байгаа кодын хэсгээс

dvrem() функц нь эгшиг гээх үүрэгтэй дүрэм юм.

```
33 references
public static string dvrem( string value, int len)
{
    return value.Remove(value.Length - 2, len);
}
```

Зураг 39. dvrem() функц

Үр дүн: Жишээ нь *хан+аа* гэсэн нөхцөлийг залгахад dvrem() функцийг дуудсанаар *хнаа* гэж залгах бөгөөд *а* гээх юм.

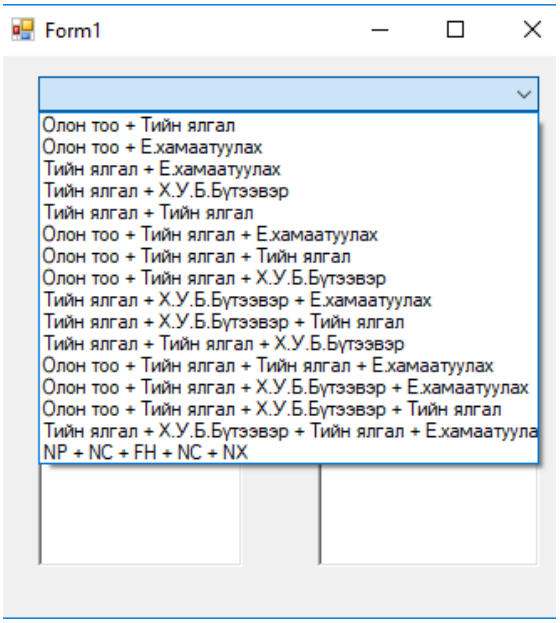
```
ret[1, k] += dvrem(NCFH1[1, i], 1) + row[j].Field<string>("n_seq_cyr");
ret[1, k + 1] += dvrem(NCFH1[1, i + 1], 1) + row[j + 1].Field<string>("n_seq_cyr");
ret[1, k + 2] += dvrem(NCFH1[1, i + 2], 1) + row[j + 2].Field<string>("n_seq_cyr");
ret[1, k + 3] += dvrem(NCFH1[1, i + 3], 1) + row[j + 3].Field<string>("n_seq_cyr");

ret[0, k] += NCFH1[0, i] + "+" + row[j].Field<string>("n_code");
ret[0, k + 1] += NCFH1[0, i + 1] + "+" + row[j + 1].Field<string>("n_code");
ret[0, k + 2] += NCFH1[0, i + 2] + "+" + row[j + 2].Field<string>("n_code");
ret[0, k + 3] += NCFH1[0, i + 3] + "+" + row[j + 3].Field<string>("n_code");
```

Зураг 40. dvrem() функцийг дуудаж хэрэглэж байгаа хэсэг

2 болон 3 давхар нөхцөлүүдийг дэд функцээр дуудаж ашиглахаар кодчилж өгсөн. Функц болгож өгсөн зорилго нь 3-тай нөхцөл дээр 2-той функцийг дуудаж ажиллуулах 4-тэй нөхцөл дээр 3-тай функцийг дуудаж ашиглах. Ингэж шийдэж өгснөөр программыг ажиллах хугацаа хэмнэх, код эмх цэгцтэй болох гэх мэт олон давуу тал бий болж байгаа юм.

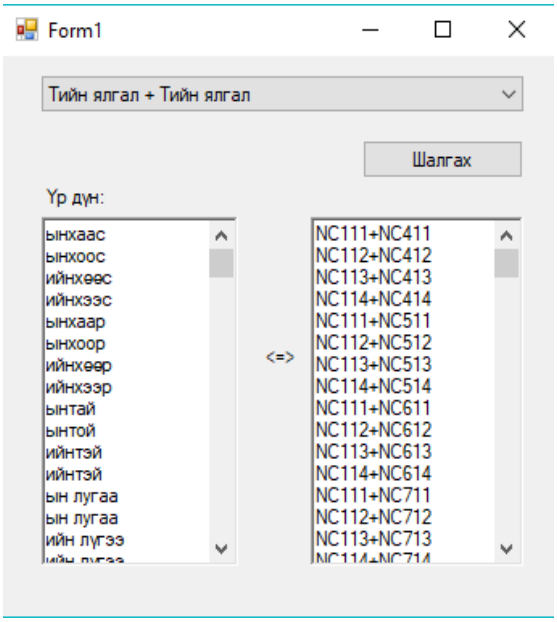
```
1 reference
public static string[,] NPNC()...

2 references
public static string[,] NCNX()...

4 references
public static string[,] NCFH()...

3 references
public static string[,] NCNC()...

2 references
public static string[,] NPNCNC()  //NP + NC + NC...

2 references
public static string[,] NPNCFH()    // NP + NC + FH...

3 references
public static string[,] NCFHNC()    // NC + FH + NC...
```

Зураг 41. 2 болон 3 давхар нөхцөлүүдийн зарим функцүүд

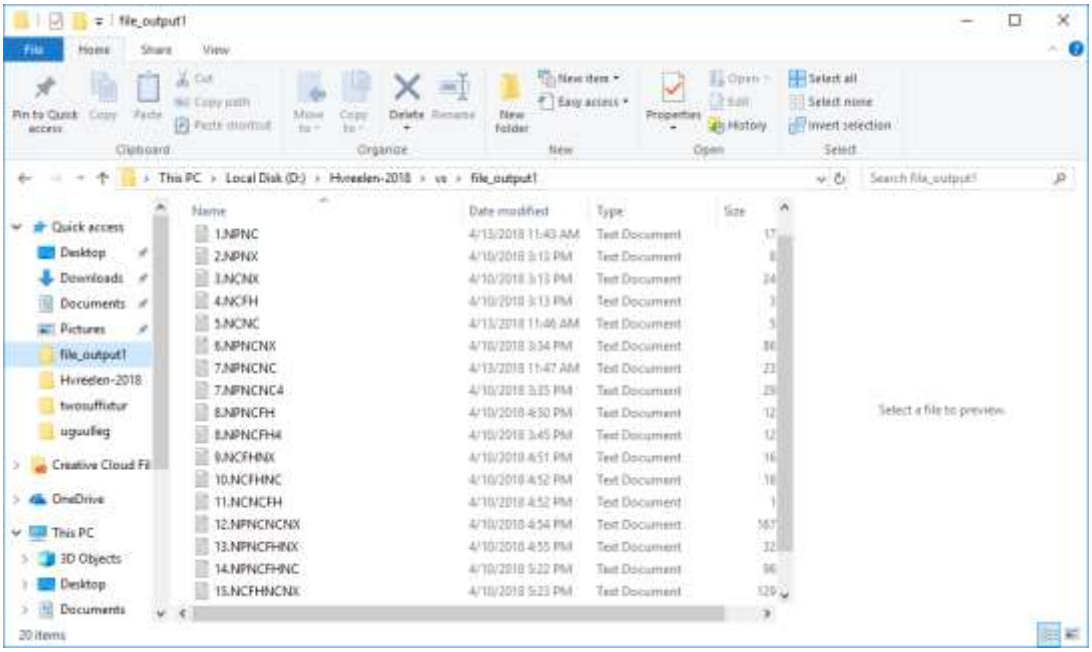


Боломжит нөхцөлийг сонгох

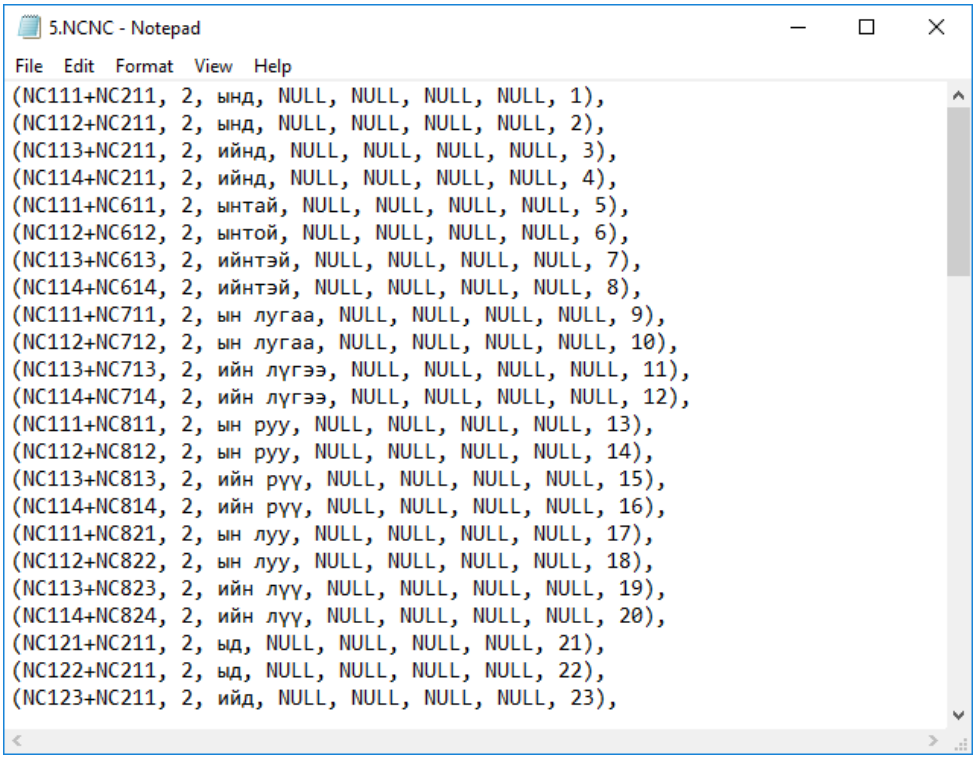
Зураг 42.



Зураг 43. Үр дүн



Зураг 44. Файл руу хадгалах



Зураг 45. Файлд хадгалсан байдал. (2 давхарласан нөхцөлийн сан)

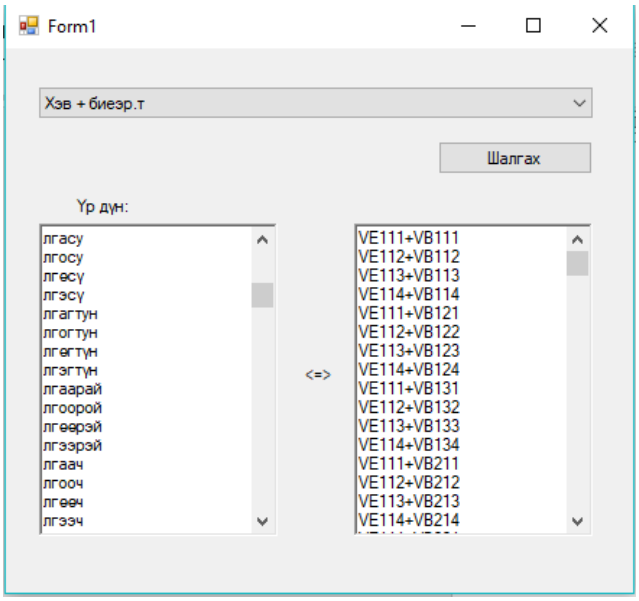
Нэр үгийн нөхцөл холбохтой адил мөн дүрмийн функцүүдийг ашигласан. Жишээ болгож кодыг хэсгийг (*зураг 46*) оруулав.

durem() функц нь лга+аарай
холбоход а–г гээж лгаарай гэсэн үр дүнг дүрмийн хувьд зөв залгах функц юм.

```
28 references
public static string durem(string value, int len)
{
    return value.Remove(value.Length - 1, len);
}
```

Зураг 46. durem() функц

Жишээ нь: **Хэв** дээр **Биеэр төгсөх** нөхцөлийг залгасан програмын үр дүн.



Зураг 47. Программын үр дүнгээс...

Ач холбогдол:

- Программаар шийдэж өгсөн учраас бүх байж болох боломжит хослолуудын үр дүнг гаргаж авна.
- Программ маш хурдан ажиллана
- Ганц удаа задалж, залгаж байгаа учир алдаа бага гарна.

Эхэд шинжилгээ хийх программ хангамж

Энэхүү программ хангамжийг хөгжүүлсэнээр бид өөрсдийн сангаа улам өргөжүүлэх ба холбогдох өгүүлбэрийн сантай болсноор үгийн утгийн алдааг олоход ашиглах явдал юм. Энэ бүтээсэн программаа бусад энэ чиглэлийн судалгаа шинжилгээ хийдэг эрдэмтэн, судлаачид, багш, оюутнуудад нээлттэй хэлбэрээр цахим орчинд байршуулахыг зорьж байна. Программ нь хэрэглэгчийн оруулсан эхийг үг болон өгүүлбэрээр задлах үүрэгтэй. Программийг хөгжүүлэх болсон зорилго нь: Үүнд:

- Кирилл, монгол бичгийн эхийг үг болон өгүүлбэрээр задлах
- Үг болон өгүүлбэрийн параллель сан үүсгэх
- Бүрдүүлсэн хөмрөгийг үгийн алдаа засах, n-gram бодоход ашиглах

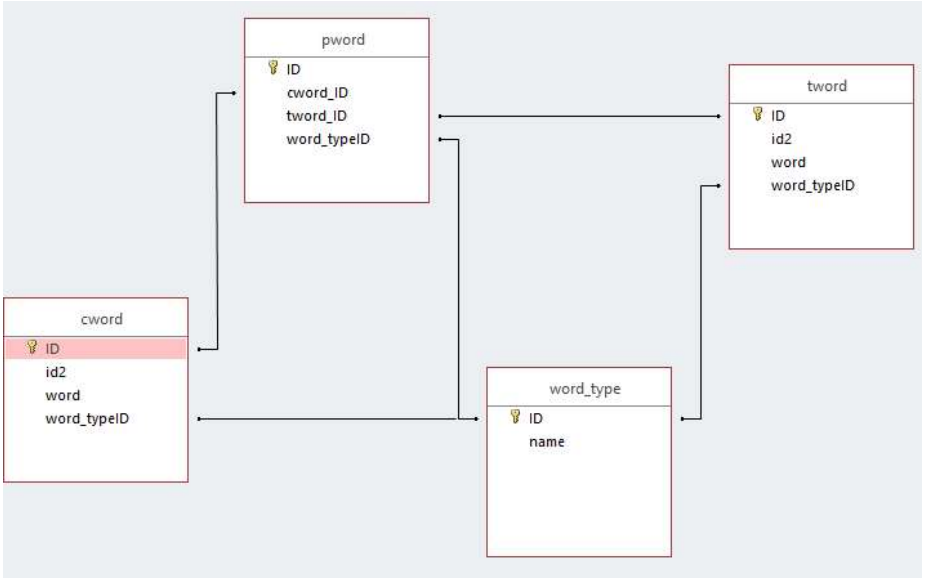
№	Тайлбар	Тэмдэг
1	Хоосон зай	“ ”
2	Цэг	“ . ”
3	Таслал	“ , ”
4	Анхаарлын тэмдэг	“ ! ”
5	Асуултын тэмдэг	“ ? ”
6	Бусад... (гэх мэт)	...

Хүснэгт 24. Үг ялгах тэмдэгт

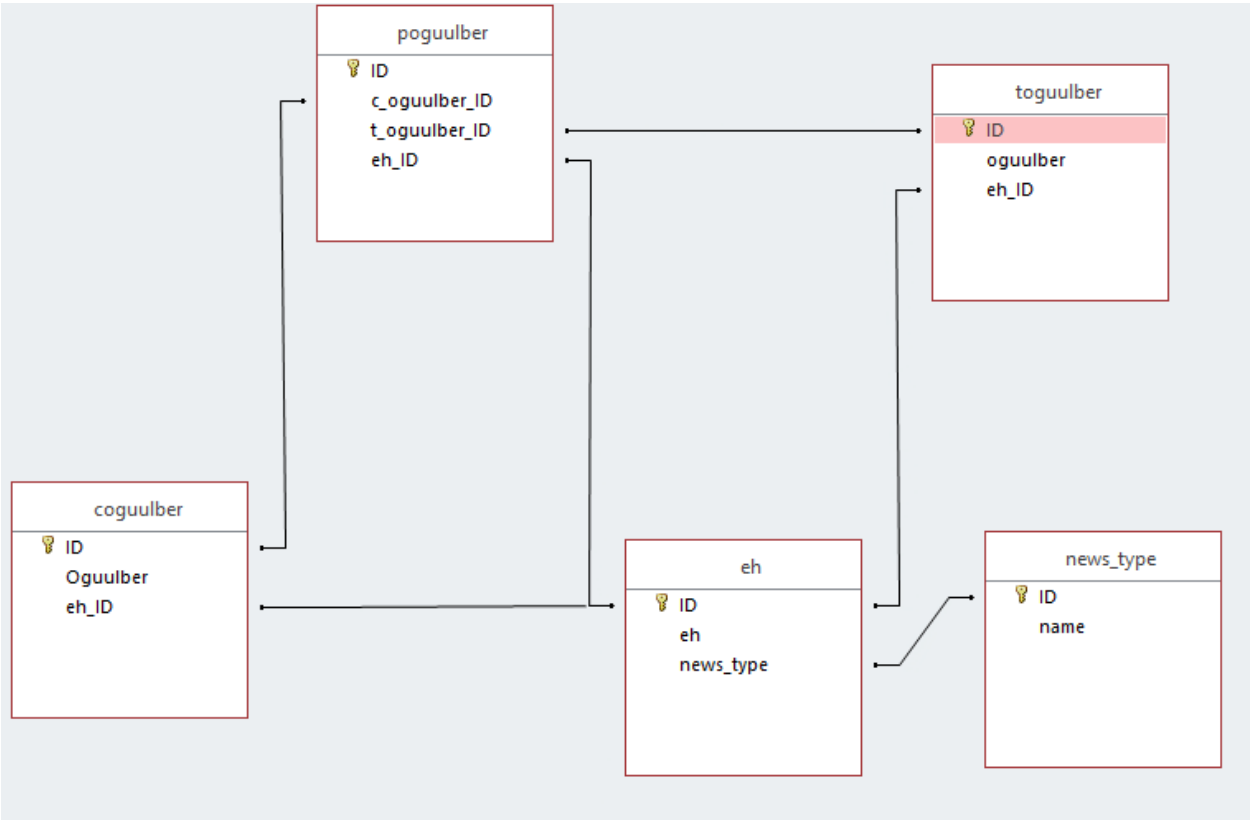
№	Тайлбар	Тэмдэг
1	Цэг	“ . ”
2	Анхаарлын тэмдэг	“ ! ”
3	Асуултын тэмдэг	“ ? ”
4	Бусад... (гэх мэт)	...

Хүснэгт 25. Өгүүлбэрийн ялгах тэмдэг

Дээрх хүснэгтэнд байгаа тэмдэг, тэмдэглэгээ болон бусад тэмдэгтүүдийг таньж салгана.



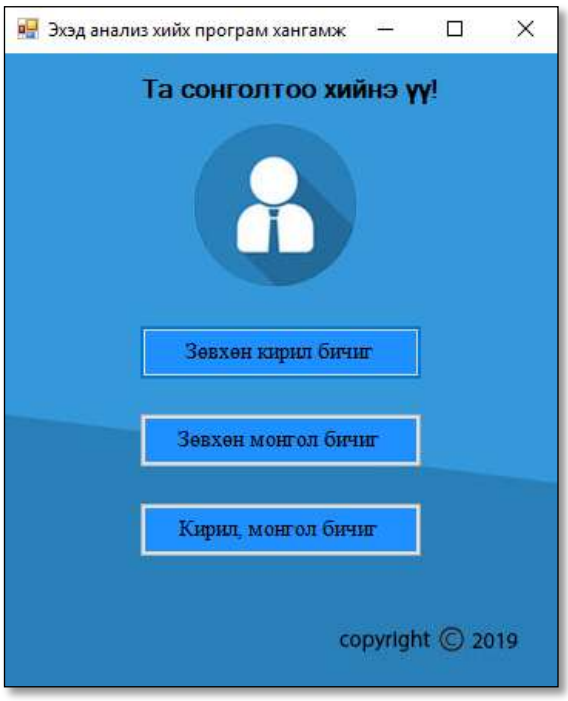
Зураг 48. Үгийн сангийн бүтэц



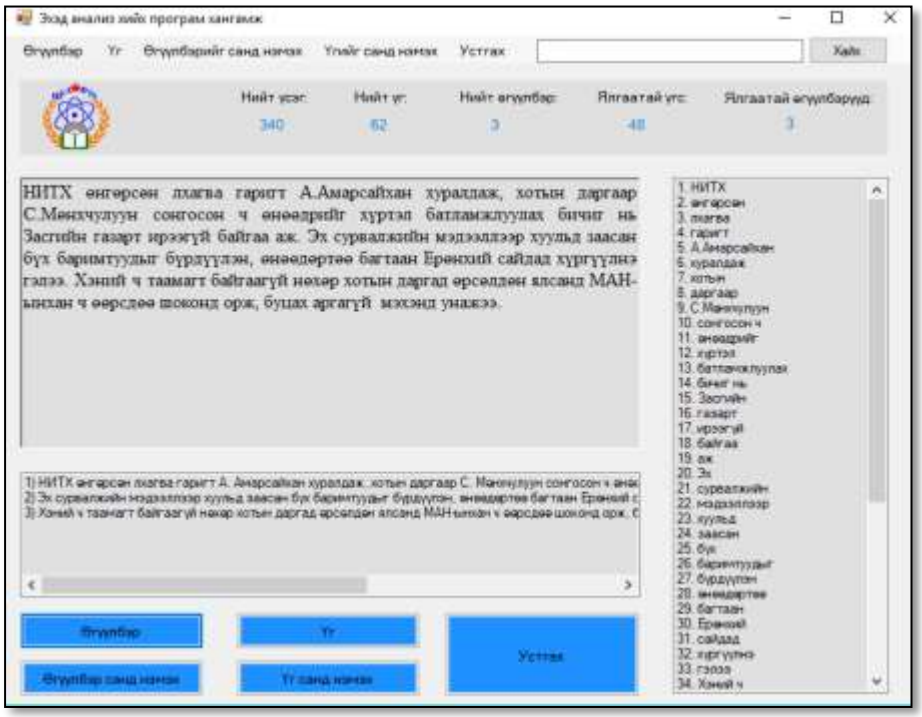
Зураг 49. Өгүүлбэрийн сангийн бүтэц

Дараах гурван үндсэн хэсгээс бүрдэнэ.

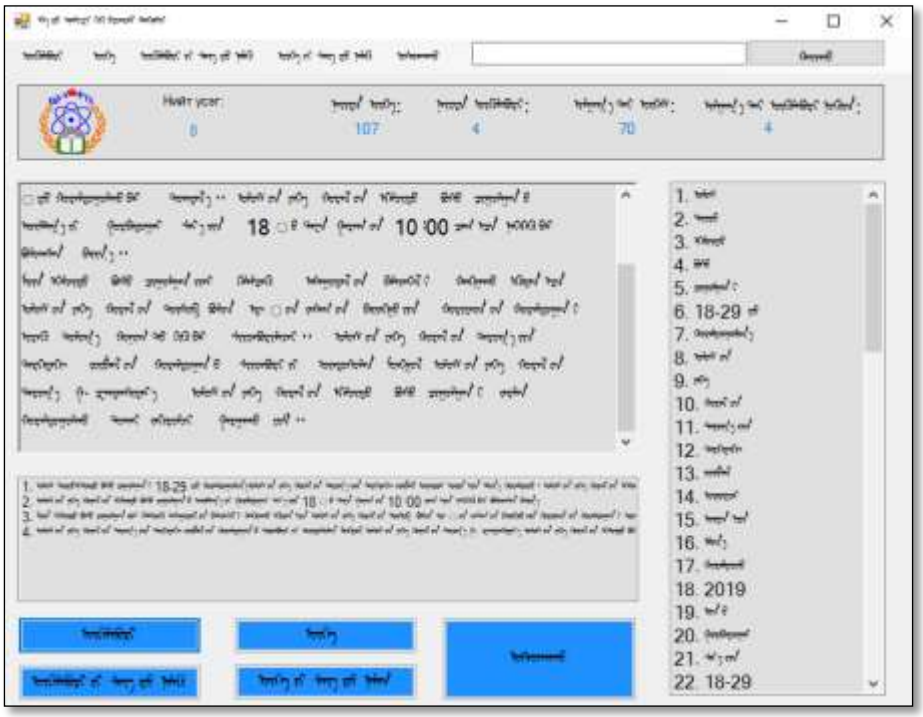
1. Зөвхөн кирилл бичиг
2. Зөвхөн монгол бичиг
3. Кирилл, монгол бичиг (параллель)



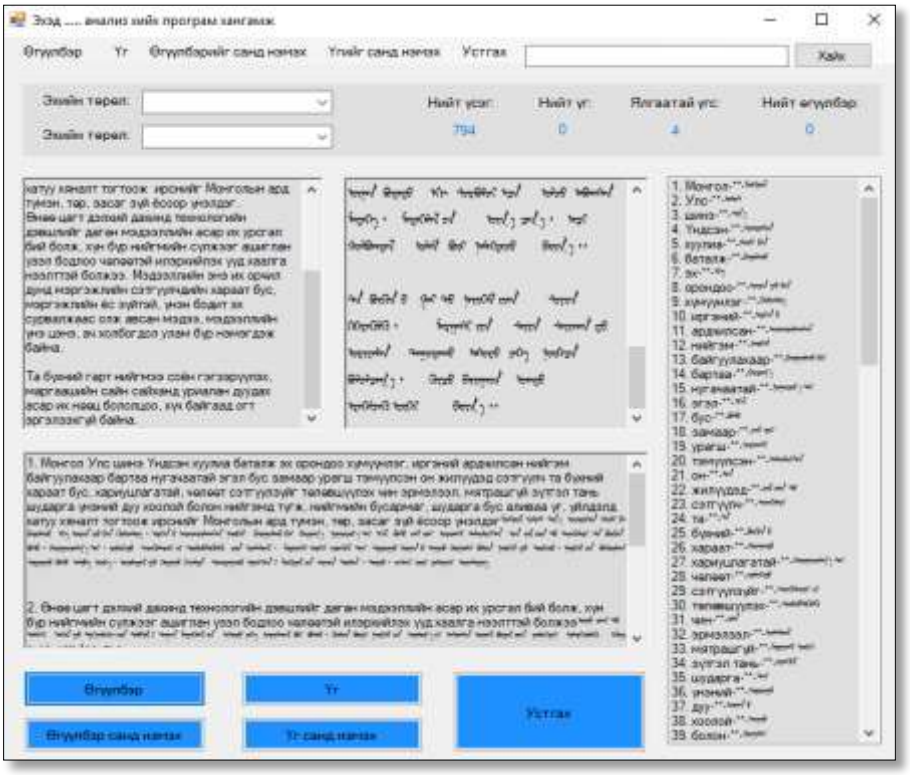
Зураг 50. Эхэд анализ хийх программ хангамжийн нүүр хэсэг



Зураг 51. Кирилл бичгийн хэсэг



Зураг 52. Монгол бичгийн хэсэг

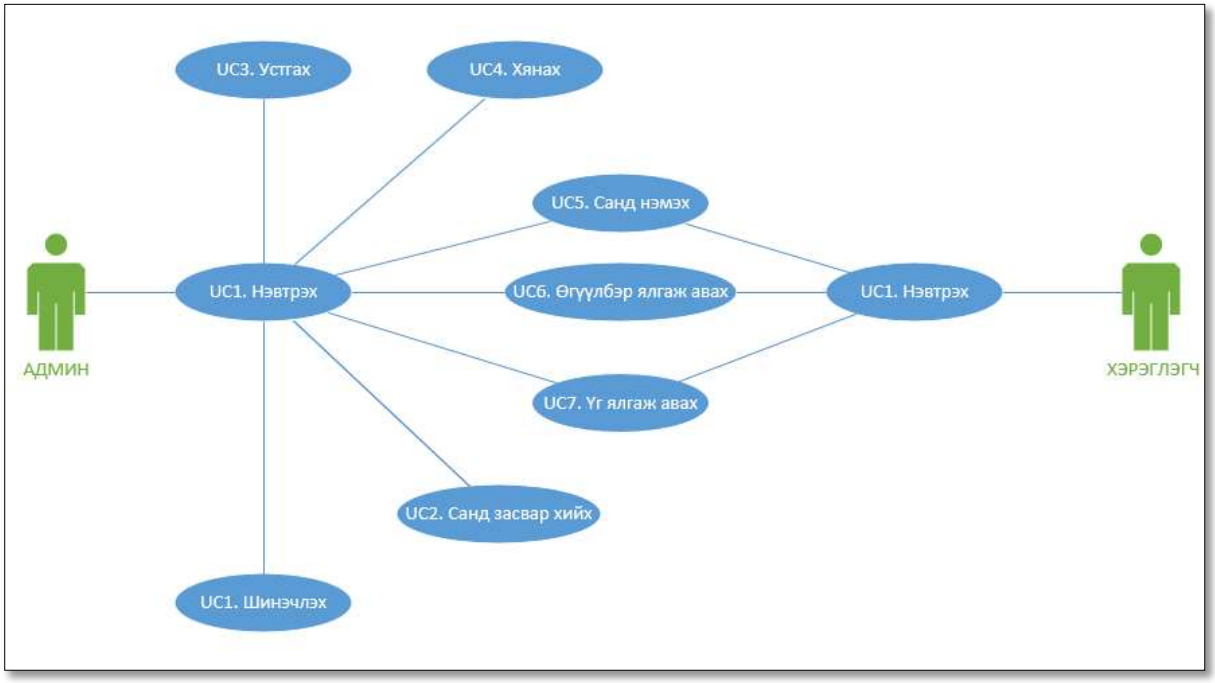


Зураг 53. Кирилл, Монгол бичгийн (параллель) хэсэг

Үг болон өгүүлбэрийг ялгахад *хүний нэр, бутархай тоо, сул үг, дугаарласан өгүүлбэрүүд* гэх мэт хүндрэлтэй учирч байсан ч программ хангамжийг хөгжүүлэх явцдаа шийдэж өгсөн. Энэхүү программ хангамжийг хөгжүүлснээр:

- Дан кирилл бичиг
- Дан монгол бичиг
- Кирилл, монгол бичиг (параллель) гэсэн 3 төрлийн үг болон өгүүлбэрийн санг бүрдүүлсэн.

Иймд бид дээрх гурван төрлийн санг ашиглан үг зүйн анализ хийх, алдаа шалгах, n-gram бодох, 2 бичгийн хооронд хөрвүүлэлт хийх зэрэг бидний ажилд ашиглагдахаас гадна хэл шинжээчид, хэл шинжлэлийн салбарын оюутнуудад ашиглаж болохуйц чухал программ хангамж болсон.



Зураг 54. Эхэд анализ хийх программ хангамжийн UC диаграмм

Бүрдүүлсэн өгөгдлийн санг удирдах системийн хөгжүүлэлт

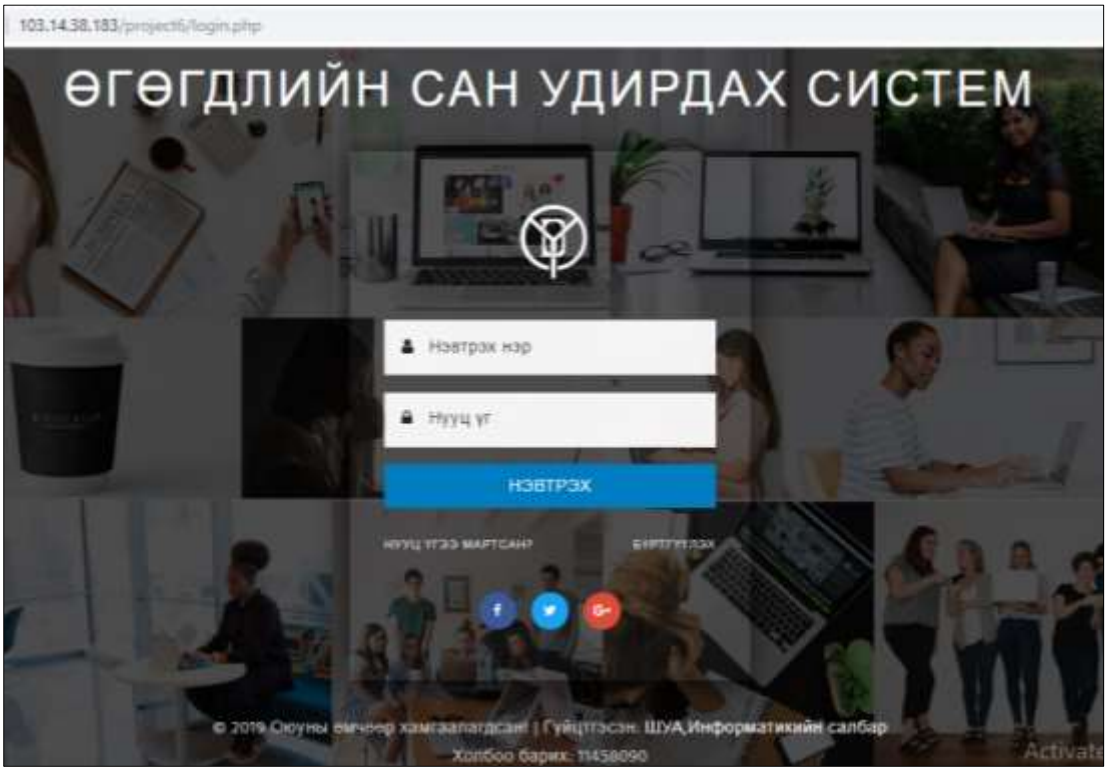
Энэхүү системийг хөгжүүлсэнээр парлаллель сан бүрдүүлэх боломжтой болсон. Систем нь хэрэглэгч, админ гэсэн 2 эрхтэй байх бөгөөд хэрэглэгч өөрий эрхээр нэвтрэн харагдаж байгаа санд зөвшөөрөгдсөн үйлдлийг гүйцэтгэх буюу кириллээс монгол бичиг рүү эвсэл монгол бичгээс кирилл рүү хөрвүүлэх зэрэг ажлыг хийнэ. Хэрэглэгч дуртай үедээ хаанаас ч хандаж ажиллах боломжтой. Хийх үйлдэл админ:

- Сан нэмэх, хасах
- Хэрэглэгч нэмэх, хасах
- Засвар хийх
- Өөрчлөлтийг хянах

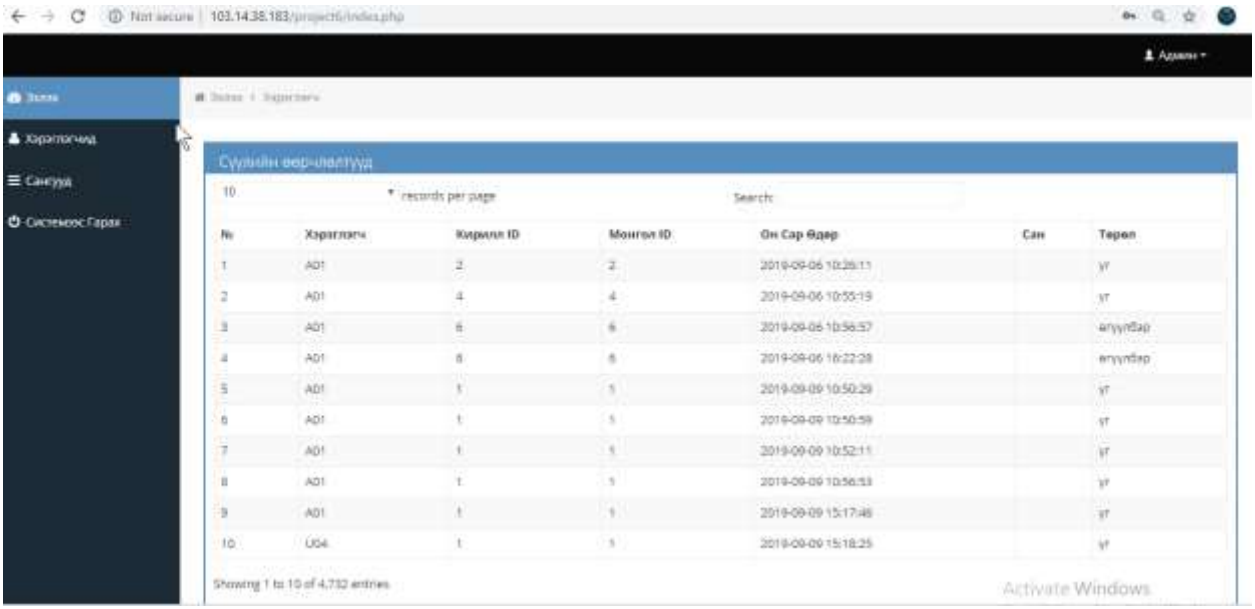
Хийх үйлдэл хэрэглэгч:

- Үг хөрвүүлэх
- Хэрэглэгчид харагдах
- Үгийг засах

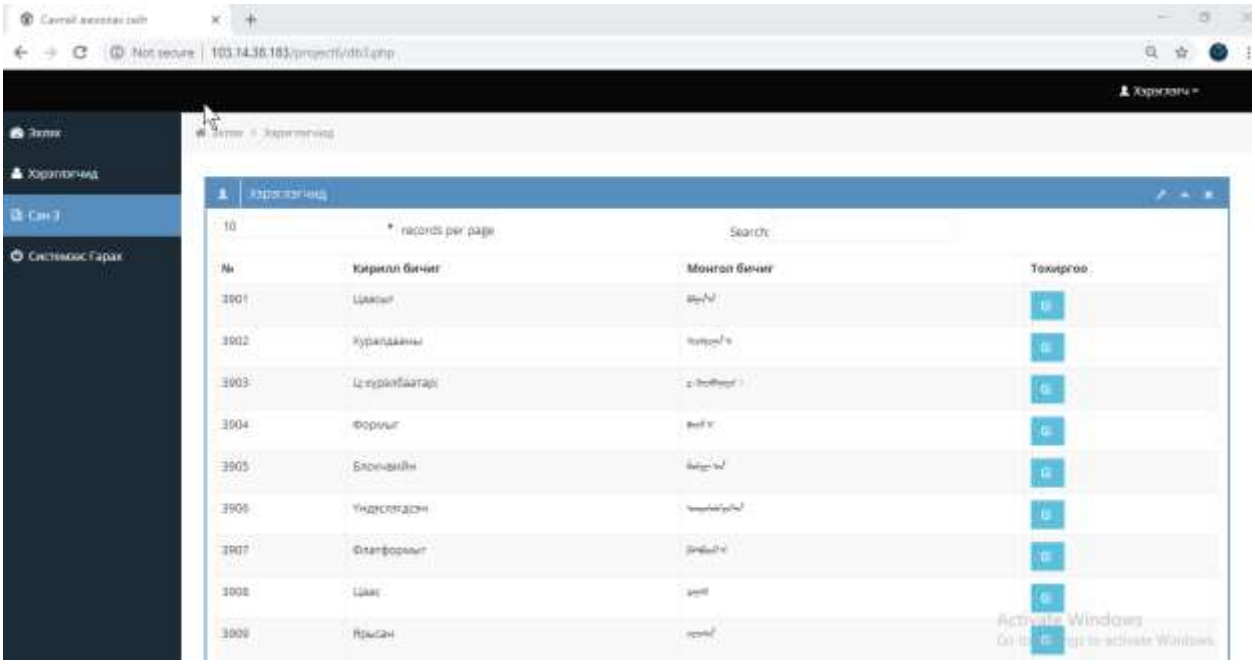
Өгөгдлийн сан удирдах системд бидний бүрдүүлсэн тайлангийн 1-р бүлэгт дурдсан өдрийн сонин, монгол толь, 108 боть зэрэг үгийн, n-gram-ын, өгүүлбэрийн сангуудыг оруулсан.



Зураг 55. Хэл боловсруулалтанд ашиглах өгөгдлийн сан удирдах системийн нэвтрэх цонх



Зураг 56. Өгөгдлийн сан удирдах систем



Зураг 57. Өгөгдлийн сан удирдах систем

Линк: http://103.14.38.183/project6/login.php

3.4 Бүлгийн дүгнэлт

Тайлангийн 3-р бүлэгт кирилл болон монгол бичгийн бичвэрийн алдааг илрүүлж засах шаардлага үндэслэл, алдааг илрүүлж засдаг алгоритмуудын судалгаа, холбогдох туршилт түүний үр дүнг авч үзсэн.

Бичвэрийн алдааг илрүүлж засах алгоритмууд, алдааны төрлүүд, үгийн алдааг илрүүлэх, засах талаар судлаж түгээмэл хэрэглэгддэг алдаа шалгуурын алгоритм болох Левенштэйн, N-Gram аргыг аргуудыг ашиглан кирилл болон монгол бичгийн бичвэрээс алдааг илрүүлж, засах туршилтыг хийсэн. Үгийн алдааг бичиглэлийн алдаа (Non-Word Error) буюу утгагүй алдаа, үг сонголтын алдаа (Real-Word Error) буюу утгатай алдаа гэж хоёр хуваан авч үзээд үгийн бичиглэлийн алдааг олохдоо Левенштэйний алгоритм буюу хоёр тэмдэгтийг ойролцоолох буюу тэмдэгтийн зөрүүг олдог алгоритмд тулгуурлан, N-Gram аргыг үгийн утгын алдааг олоход ашиглаж холбогдох программын кодыг бичин туршилтыг хийлээ. N-Gram аргыг Back-off soomthing, Sparse Matrix, MLE (Maximum Likelihood Estimation), MED (Minimum Edict Distance) зэрэг аргуудын хамт ашигласан.

Ингэж үгийн алдааг шалгах алгоритмууд, алдааг олох арга зүйн талаар судлаж хамгийн тохиромжтой, түгээмэл хэрэглэдэг алгоритмыг сонгон авч холбогдох программ хангамжуудыг бичин туршихад дараах дүгнэлтүүд гарч байна.

1. Тайлангийн 3-р бүлгийн туршилтуудаараа программ хангамжийн ажиллагаа болон бусад зүйлийн талаар ойлгомжтой харагдуулахыг зорилоо. Тухайлбал бүрдүүлсэн сан дээрээ тулгуурлан эхний оруулсан нийтлэл (эх) дээрээ 3 янзаар туршилт хийж үзлээ.
   - Туршилтанд оруулсан эхээс санд байхгүй үг 293, алдаатай үг 101 байсан бөгөөд энэхүү нийтлэлийн алдаатай үгсийг засаж, санд нэмж дараагийн туршилтыг хийж үзэв.
   - Дахин шалгаж үзэхэд 4 ширхэг үг алдаатай гарч ирсэн. Тухайн үг нь зөв бичих дүрмийн алдаагүй харин өөр тэмдэгтийн хамт байсан учир үүнийг программ алдаатай гэж үзсэн. Өгүүлбэрээс үгийг салгаж авахдаа энэхүү асуудлыг программын код дээрээ шийдэж өгсөн.
   - Албаар алдаатай 11 ширхэг үгийг бичиж өгөөд шалгахад 11 үгийг бүгдийг нь алдаатай байна гэж илрүүлсэн.
   - *N-gram* арга нь эхлээд trigram бодоод олдохгүй бол bigram бодоод олдохгүй бол monogram бодож үгийн алдааг илрүүлж засах үгийг санал болгох ба алдаатай бичигдсэн үгийг олох боломжтой гэдэг нь туршилтаас харагдсан.
2. Кирилл болон монгол бичгийн бичвэрээс үгийн алдаа шалгаж илрүүлэхэд энэ төрлийн алгоритмуудаас Левенштэйний алгоритмын хамт N-gram аргыг ашиглахад хамгийн үр дүнтэй болох нь судалгааны үр дүн болон туршилтаас харагдсан.
3. Туршилт үр дүнгийн харьцуулалтаас харахад Левенштэйний алгоритмыг дангаар нь ашиглахад зөвхөн тухайн үгийн алдаатай эсхийг шалгаж байгаа бол N-gram аргыг хамт ашигласнаар утгын алдаа, үг сонголтын алдааг илрүүлэн санал болгож байгаа нь давуу тал болж өгч байна. Өвөр Монголын Их Сургуулийн хийсэн алдаа шалгуур дээр шалгаж үзэхэд алдаа үгийг засаж мөн утгын алдааг тодорхой хэмжээнд илрүүлж байна.

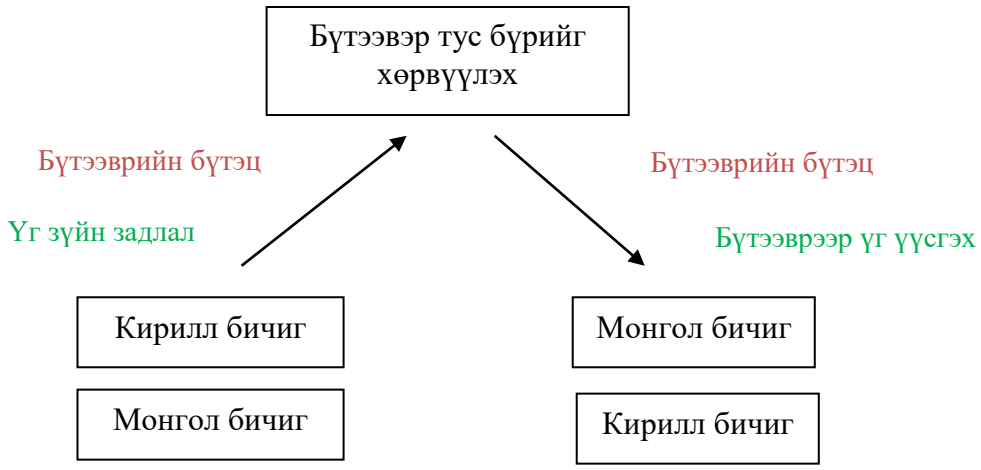
4. Үгийн алдаа шалгуурын программ хангамжийг хөгжүүлж, тодорхой туршилт хийхийн тулд нөхцөлийн дарааллийн сан бүрдүүлэх, эхэд анализ хийх, бичвэрийн сан бүрдүүлэх зэрэг холбогдох зарим программ хангамжийг хөгжүүлж ашигласан. Программ хангамжийн үр дүн нь ямар текст оруулж байгаагаас болон үгийн санд хэдэн үг байгаасаа шууд хамааралтай байгаа учраас программын үр дүнг сайжруулахын тулд өгөгдлийн сангаа маш сайн өргөжүүлэх шаардлагатай байна. Энэ судалгааны ажил нь компьютер хэл шинжлэлийн чиглэлээр мэргэших программ хангамжийн оюутанууд, хэл шинжлэлийн салбарын оюутнуудын судалгаанд тус нэмэр болно. Цаашид төслийн багийн хамт олон илүү их сан бүрдүүлж, бичвэрээс алдаа илрүүлэх, засах программаа сайжруулж хэрэглээнд нэвтрүүлэхийг зорьж байна.
5. Өгөгдлийн сангаа үгийг үндсээр нь бүрдүүлэх учир дүрэм шинээр өөрчлөгдөхөд бүрэн нийцэж шинэчлэгдэх боломжтой. Мөн алдааг шалгахаас гадна хамгийн оновчтой хувилбарыг санал болгох боломжийг бүрдүүлэх нь чухлаар тавигдана.
6. Мэргэжлийн нэр томъёо, нийгмийн олон талт хөгжлийг даган гарч буй шинэ үгийг утгын тайлбар, бусад холбогдох мэдээллийг санд нэмэн оруулах бүрэн боломжтойгоос гадна хэрэглэгчид үгсийн санг нэмэн дэлгэрүүлэх, холбогдох мэдээллийг оруулах, санал болгох боломжийг олгох хэрэгтэй.

IV. КИРИЛЛ БОЛОН МОНГОЛ БИЧГИЙН БИЧВЭР ХООРОНД ХӨРВҮҮЛЭХ ЗАГВАР

4.1 Дүрэмд суурилсан арга

Энэхүү хоёр бичиг хооронд хөрвүүлэх гэдэг нь нэг бичгээр өгөгдсөн үгийг бүтцээр задалж, үгийн үндэс болон нөхцөлүүдийг олно. Ингээд өгөгдлийн сантай харьцуулж үгийн үндэс болон нөхцөлүүдийн нөгөө бичгээрх хэлбэрийг тодорхойлно. Эцэст нь олж тодорхойлсон үндсийг өгөгдсөн нөхцөлүүдээр хувилгах замаар нөгөө бичигт хөрвүүлнэ [19]. Хөрвүүлэх ерөнхий зарчмыг дараах зурагт тодорхойлов.

Монгол хэлний кирилл-монгол бичгийн хөрвүүлгийн системийн алгоритм

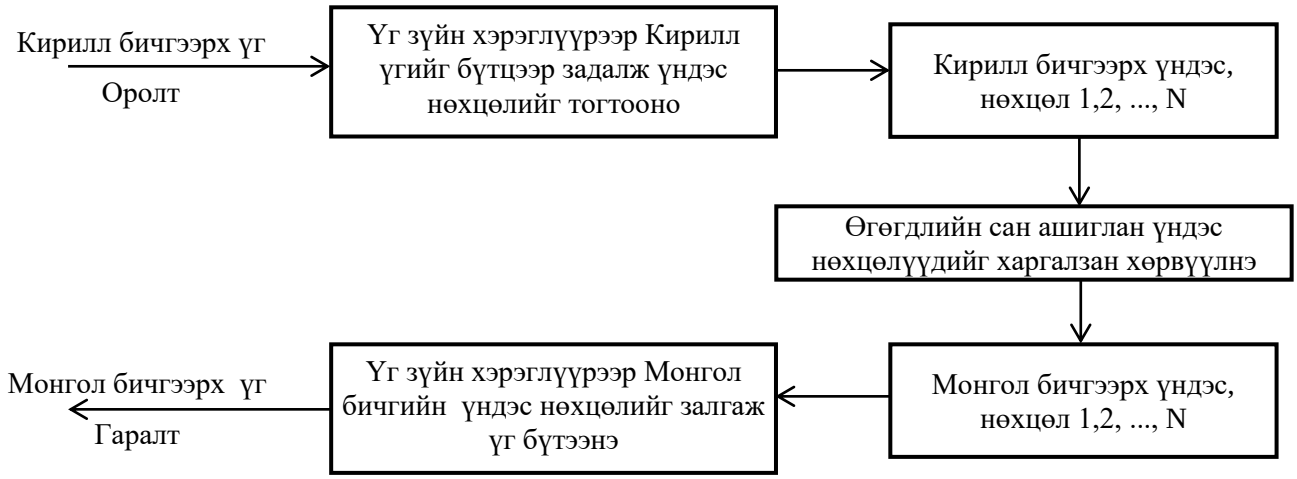


Зураг 58. Хөрвүүлэх ерөнхий зарчим

Жишээлбэл,
аавын → аав + ын → аав + NC2 (харьяалахын т.я)
abu + NC2 → abu + iian → abu-iian

Одоо хөрвүүлэлт тус бүрийн онцлогт нь тохирсон схемийг авч үзье.

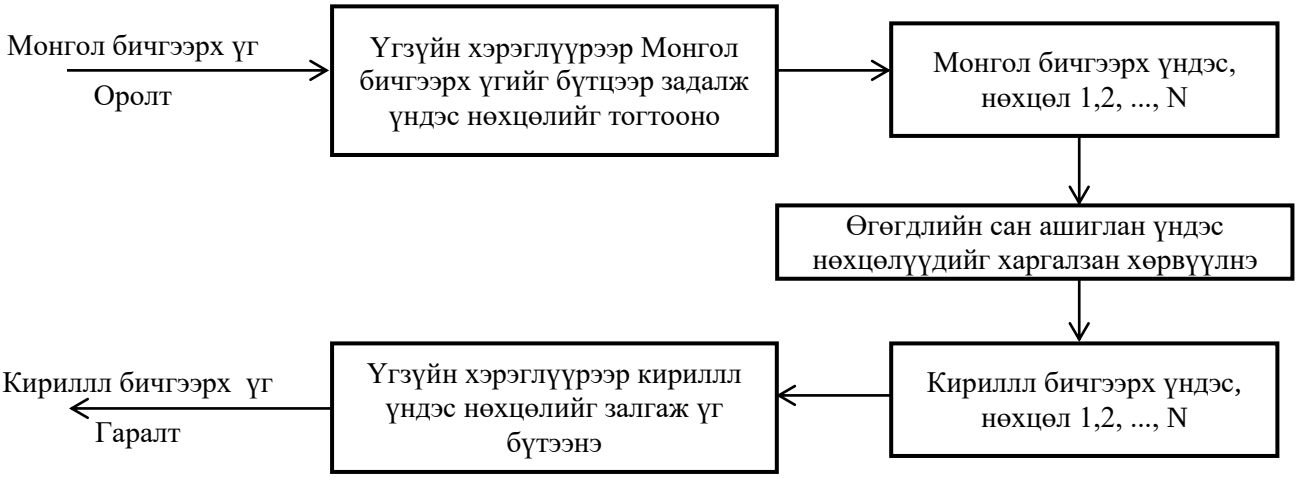
1. Кирилл бичгээр бичсэн үгийг монгол бичигт хөрвүүлэх



Зураг 59. Кирилл бичгээр бичсэн үгийг монгол бичигт хөрвүүлэх загвар

Кирилл бичгээс монгол бичигт хөрвүүлэх үед кирилл үг нь олон хэлбэрээр бүтцээр задрах болон үгийн утгаас хамаарч монгол бичгээр ялгаатай бичигдэх боломжтой байдаг. Иймээс үгийн утга таних асуудал чухлаар тавигдана.

2. Монгол бичгээр бичсэн үгийг кирилл бичигт хөрвүүлэх



Зураг 60. Монгол бичгээр бичсэн үгийг кирилл бичигт хөрвүүлэх загвар

Монгол бичгээс кирилл бичгээс хөрвүүлэх үед монгол бичгээр бичигдсэн оноосон нэрийг ялгаж таних асуудал чухлаар тавигдаж байна.

- *Хөрвүүлэх бичвэрийг өгүүлбэр задлуураар өгүүлбэрт задална*.

Өгүүлбэрийн төгсгөлийг монгол хэлний зөв бичих дүрмийн дагуу цэг, таслал, асуултын тэмдэг, анхаарлын тэмдэг зэрэг өгүүлбэрийн төгсгөл заагчаар таньж болно. Харин өгүүлбэр дунд орсон товчлолын ард бичсэн цэг (жишээ нь, ам. доллар, док. Н.Баянмөнх), гишүүн өгүүлбэр төгсгөсөн цэг, таслал, асуултын тэмдэг, анхаарлын тэмдэг зэргийн араар өгүүлбэр төгсөж байна гэж үзэж болохгүй.

- *Өгүүлбэрийг token зааглагч буюу үгээр задална.*

Монгол хэлний зөв бичих дүрмийн дагуу өгүүлбэрийн утгат, нэг бүхэл (atomic) хэсгүүдийг тодорхойлох үйлдэл юм. Утгат хэсэг буюу үгээр салгахдаа зураастай үг (и-мэйл, Хан-Уул), хүний нэрийн товчлол (В.И.Ленин, Батж.Батбаяр), зэрэг цолны товчлол (проф., др., маг.), бусад товчлол (м/с, ам.), латин үсгээр бичигдсэн үг (MS Word, Android), огноо (2019.06.13, 12-01-30, 86/02/09), цаг (15:59), тоо (0.15, 33, 15,500), и-мэйл эсвэл URL (info@mas.ac.mn, http://www.mas.ac.mn) зэргийг халгалзан үзэх шаардлагатай.

- *Үгийг үг зүйн загварчлалын программаар* ***үндэс+нөхцөл*** *болгон бүтээврээр задална.*

```
ботиуд = боть + ууд (боть + NP2)
уушгиар = уушги + аар (уушги + NC5)
мориноос = морь + ооc (морь + NC5)
```

- *Бүтээвэр тус бүрийг кирилл болон монгол хөмрөгөөс харгалзуулан хөрвүүлнэ.*

Бүтээвэр тус бүрийг үндэс, нөхцөлийн хөмрөг буюу үгийн жагсаалтаас хайх буюу Dictionary lookup аргыг ашиглан хөрвүүлнэ.

```
боть + NP2  ->    boti + NP2
уушги + NC5 ->    ayuski + NC5
```

```
морь + NC5  ->    mori + NC5
```

- *Бүтээврээр залган хөрвүүлсэн үгээ үүсгэнэ.*

```
boti + NP2 = boti + uud
ayuski + NC5 = ayuski + aca
mori + NC5 = ayuski + aca
```

Ингээд туршилт хийсэн өгөгдөлдөө дүн шинжилгээ хийж үзсэн.

1. Кириллээс монгол бичигт хөрвүүлэхэд:

Туршилт хийсэн бичвэрийн зөв бичих дүрмийн алдаанаас болж хөрвүүлээгүй буюу буруу хөрвүүлсөн үг 4.6%, 2 болон түүнээс дээш хувилбараар задалсан үгээс зөв задалсан хувилбарыг сонгохдоо алдсан алдаа 1.9% , үгийн утга таних аргуудын туршилтанд оруулаагүй үгүүдийн хөрвүүлэхдээ алдсан алдаа 6.7%-ийг эзэлж байна. Харин тусгай сан болон холбоо үгийн санд оруулж утга таних туршилт хийсэн үгүүдийн хувьд 98%-ийн үр дүнтэйгээр зөв хөрвүүлсэн байлаа.

2. Монгол бичгээс кирилл бичигт хөрвүүлэхэд:

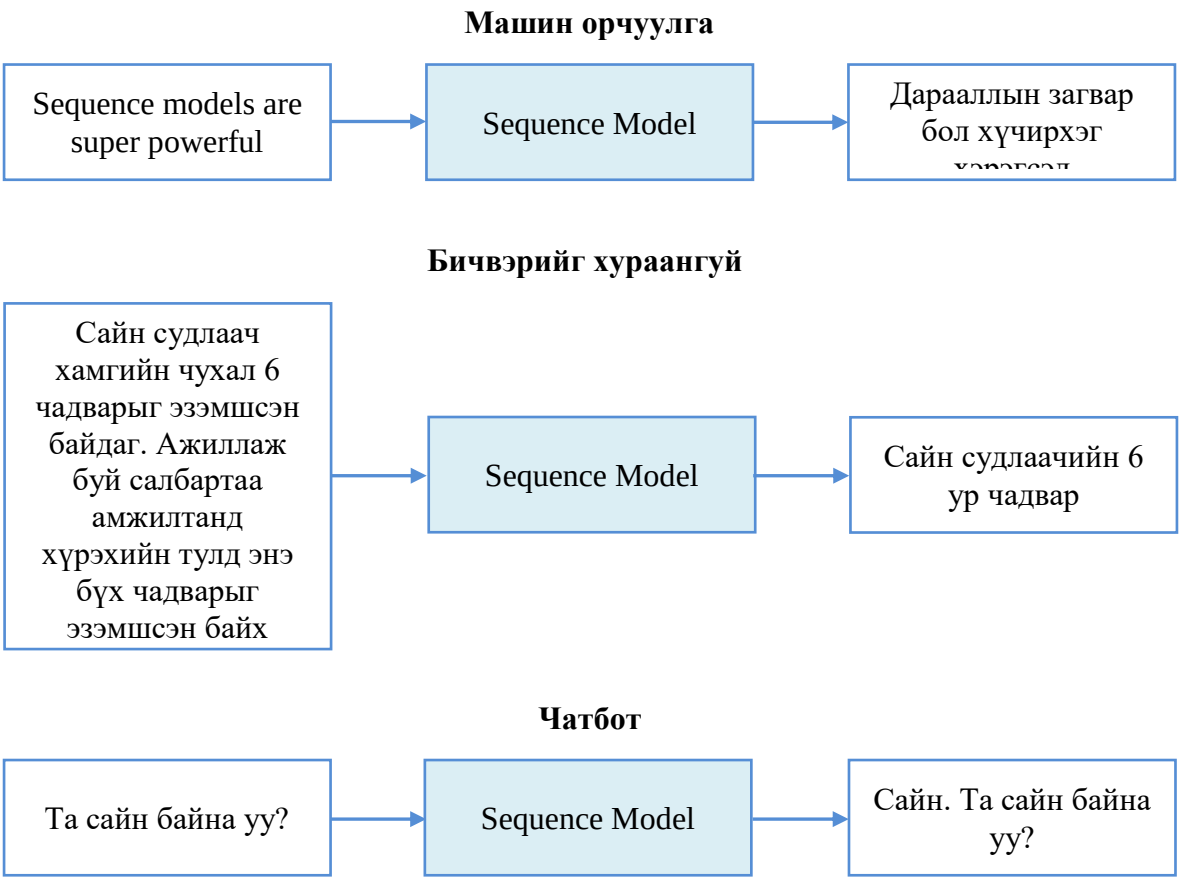
Туршилт хийсэн бичвэрийн зөв бичих дүрмийн алдаанаас болж хөрвүүлээгүй буюу буруу хөрвүүлсэн үг 8.6%, үгийн утга танихтай холбоотой алдаа 5.7%-ийг эзэлж байна.

Үүний дараа хоёр бичвэрийн туршилт хийсэн өгөдлүүдийг зөв бичих дүрмийн дагуу нэг бүрчлэн засаж дахин хөрвүүлэлт хийлээ. Туршилтийн үр дүн дараах байдлаар өөрчлөгдлөө.

- Кирилл бичвэрээс монгол бичгийн бичвэрт хөрвүүлэхэд 91.3%.
- Монгол бичгийн бичвэрээс кирилл бичвэрт хөрвүүлэхэд 89.1%.

4.2 Машин сургалтын арга

Их хэмжээний өгөгдөл дээр дараалалтай холбоотой асуудлыг шийдэхэд RNN (Recurrent Neural Network) хамгийн тохиромжтой машин сургалтын арга юм. Энэ арга нь яриа таних, эх хэлний боловсруулалт (NLP), хугацаанаас хамаарсан таамаглал дэвшүүлэх зэрэг олон төрлийн хэрэглээтэй байдаг. RNN архитектурын тусгай арга болох Sequence to Sequence (seq2seq) загварыг машин орчуулга, асуултанд хариулах, чатбот үүсгэх, бичвэрийг хураангуй болгох гэх мэт хэлний нарийн төвөгтэй асуудлыг шийдвэрлэхэд ихэвчлэн ашигладаг.



Зураг 61. Seq2seq моделийн хэрэглээ

Машин орчуулга

Орчуулга хийх нь зөвхөн хүний хувьд биш, бас машины хувьд ч гэсэн хэцүү даалгавар юм. Машин орчуулгын системүүд хөгжлийн эхэнд үедээ **дүрэмд суурилсан машин орчуулга (RBMT)**-ын арга хэрэглэдэг байсан бөгөөд хэл шинжээчид үг зүй, өгүүлбэр зүй, утга зүйн түвшинд орчуулга хийх дүрмүүдийг тодорхойлдлог байв. Дараа нь их хэмжээний зэрэгцээ өгүүлбэр бүхий хөмрөгөөс статистик моделийн тусламжтай хэрхэн орчуулахыг сурдаг **статистик машин орчуулга (SMT)**-ын арга хэрэглэх болов. Жишээ нь, Франц хэл (f) дээрх бичвэрийг Англи хэл (e) рүү орчуулах моделийг **(translation model)** $p(f|e)$ гэе. Энэ моделийг хоёр хэлний зэрэгцээ өгүүлбэр бүхий хөмрөг дээр сургах бөгөөд хэлний моделийг **(language model)** зөвхөн зорилтот хөмрөг (Англи хэлний хөмрөг) дээр тооцоолдог.

$$\boldsymbol{e = \text{argmax}_e p(e|f) = \text{argmax}_e p(f|e)p(e)}$$

Мөн (SMT) болон (RBMT) хослуулсан эрлийз систем (Hybrid System)-ийг бий болгон хэрэглэдэг болсон.

Дүрэмд суурилсан машин орчуулга (RBMT)

RBMT нь олон хэл шинжлэлийн дүрэм, олон сая хос хэлний толь бичиг дээр үндэслэдэг. RBMT технологи нь текстийг боловсруулж, орчуулж буй хэл дээр текст үүсгэх шилжилтийн дүрслэлийг үүсгэдэг. Энэ үйл явц нь морфологи, синтакт, семантик, гэх зэрэг олон тооны дүрэмтэй, их хэмжээний үгсийн санг шаарддаг. Программ хангамж

нь эдгээр нарийн төвөгтэй дүрмийн багцыг ашигладаг бөгөөд эх хэл дээрх дүрмийн бүтцийг орчуулж буй хэл рүү шилжүүлдэг.

Дүрэмд суурилсан машин орчуулгын систем нь их хэмжээний толь бичиг, хэлний маш нарийн дүрэмд тулгуурласан байдаг. Хэрэглэгчид мэргэжлийн хэллэг болон нэр томъёог толь бичгүүдэд нэмж үүсгэх замаар системийн орчуулгын анхдагч тохиргоог дарангуйлан хэрэглэж орчуулгын чанарыг сайжруулах боломжтой.

Статистик Машин Орчуулга (SMT)

Статистик машин орчуулгийн арга нь нэг болон хос хэлний сургалтын өгөгдөлд дүн шинжилгээ хийдэг статистик орчуулгын загваруудыг ашигладаг. Үнэн хэрэгтээ энэ арга нь тооцооллын хүчин чадлыг ашиглан боловсронгуй өгөгдлийн загваруудыг бий болгохын тулд нэг хэлийг өөр хэл рүү хөрвүүлдэг. Орчуулга нь алгоритм ашиглан сургалтын өгөгдлүүдээс хамгийн түгээмэл тохиолддог үг, хэллэгийг сонгодог.

SMT загварыг бүтээхэд системийг сургахын тулд хосолсон хэл болон домайныг байршуулах замаар хийгддэг хурдан бөгөөд энгийн процесс юм. Онцгой домэйнтэй системийг сургахын тулд дор хаяж 2 сая үг шаардагдах боловч үүнээс бага хэмжээтэй байж ч болох юм. SMT технологи нь хос хэлний корпус дээр суурилдаг ба толь бичиг болон орчуулгын санах ойг судалж хэлний бүтцэд сургадаг. Эмнэлэг, санхүү, техникийн гэх мэт тусгай домайн бүхий өгөгдлийг ашиглан сургасан тохиолдолд SMT моделиуд илүү сайн орчуулах боломжтой болно.

SMT технологи нь CPU-ний чадал, орчуулгатай ажиллахын тулд өргөн хүрээний тоног төхөөрөмжийн тохиргоо шаарддаг. Ийм учраас хэрэглэгч программ хангамж болон техник хангамжид их хэмжээний зардал гаргадаггүй үүлэн тооцооллын суурь системүүд илүүтэйгээр давуу талтай юм.

Дүрэмд суурилсан болон Статистик машин орчуулга (RBMT vs SMT)

- Дүрэмд суурилсан машин орчуулгын арга нь маш сайн үр дүнг гаргаж чадах боловч сургалтын болон хөгжүүлэлтийн зардал дэндүү өндөр юм.
- Дүрэмд суурилсан машин орчуулгын арга нь статистик машин орчуулгын аргыг бодвол толь бичиг болон хэлний дүрмүүдийг ашигладаггүй ба илүү бага хэмжээний өгөгдөл ашигладаг байна. Энэ нь зарим тохиолдолд үр дүнг бас бууруулдаг.
- Хэл гэдэг байнга өөрчлөгдөж байдаг. Тиймээс хэлний дүрмүүд шинэчлэгдэхэд дүрэмд суурилсан аргыг тохируулан өөрчилж байх хэрэгтэй.
- Статистик машин орчуулгын аргыг маш богино хугацаанд бүтээж болох бөгөөд системд дүрмүүдийг оруулахын тулд хэлний мэргэжилтүүд шаардлагагүй.
- Статистик машин орчуулгын загварууд нь их хэмжээний орчуулгын загваруудыг удирдан зохицуулдаг бөгөөд орчин үеийн компьютерын боловсруулах хүчин чадал, мөн өгөгдөл хадгалах багтаамжийг шаарддаг.
- Статистик машин орчуулгын системүүд нь сургалтын өгөгдлийн загварыг дууриалгаж чаддаг ба тэдгээрийн илүү давтамжтай гаралтыг бий болгох боломжийг олгодог.

Хэдийгээр IBM, Google зэрэг том компаниуд статистик машин орчуулгын системийг олон жилийн турш хөгжүүлж, хэрэглээнд нэвтрүүлэн арилжаалж байсан боловч, энэ арга нь орчуулж буй бичвэрийн хэллэг дээр хэт төвлөрснөөс орчуулгын утга санааг алдагдахад хүргэсэн.

Өмнөх үеийн, хэлцэд суурилсан уламжлалт орчуулгын системүүд өгүүлбэрийг олон дэд хэсэгт хуваагаад, тэдгээрийг хэлц тус бүрээр нь орчуулаад буцааж нэг өгүүлбэр болгодог байв. Энэ чиг хандлага нь зарим үед буруу орчуулдаг юм.

Нейрон машин орчуулга (NMT) анх 2014 онд хэрэглээнд нэвтэрсэн бөгөөд машин орчуулгын статистик моделийг сургахын тулд нейрон сүлжээ (мэдрэлийн сүлжээ) ашигласан байдаг. Энэ цаг үеэс хойш нөхцөлт магадлалыг тооцоолох шаардлагагүй болсон. Харин хэл шинжээч, статистикчид маш их энерги, хүч зарцуулж байж олж мэддэг байсан дүрэм, магадлал, жинг нейрон сүлжээнүүд өөрсдөө сурч чаддаг болсон.

RNN сүлжээ, ялангуяа LSTM-үүд дарааллын өмнөх буюу $t-1$ алхам дахь элемент дээр үндэслэн t алхмын элементийг таамагладаг. Тэгэхээр RNN сүлжээ хэлний орчуулга хийхэд туслаж чадах болов уу?

LSTM ашиглан үгсийн дарааллыг нэг хэлнээс нөгөө хэл рүү буулгах нь шууд бэрхшээлтэй тулгарна. Нэг LSTM ашиглаж байгаа тохиолдолд оролтын болон гаралтын дараалал ижил урттай байх шаардлагатай. Орчуулгын хувьд энэ нь боломжгүй юм. Жишээ нь, “is playing” гэдэг хоёр үгтэй Англи хэлцийг Герман руу орчуулахад “spielt” гэсэн нэг үгтэй хэлц болно.

Seq2Seq модель

Өмнө дурьдсан бэрхшээлтэй асуудлын шийдэл нь энкодер–дэкодер (encoder–decoder) архитектур бөгөөд **sequence-to-sequence (seq2seq)** модель гэж нэрлэдэг. Энэ аргын гол чадвар нь бичвэрийг кодлохдоо тогтмол урттай дүрслэлээр илэрхийлэх бөгөөд үүнийг контекст вектор гэж нэрлэдэг. Кодолсны дараа, уг контекстийг өөр код тайлагчийг ашиглан өөр хэл рүү орчуулдаг. Seq2Seq моделийг (Sutskever et al., 2014, Cho et al., 2014) анх Google компани машин орчуулгын хувьд танилцуулсан бөгөөд энэ нь гүний сургалт ашиглан орчуулгын үйл явцад хувьсгал хийсэн. Орчуулахдаа оролтын үгийг харгалзан үзээд зогсохгүй зэрэгцээ үгнүүдийг нь хүртэл харгалзан үздэг. Энэ моделийг ашиглан машин орчуулга, яриаг таних, текстийг хураангуйлах зэрэг олон ажил үр дүнд хүрсэн байдаг.

Өнөө үед үүнийг дүрс тайлбар, харилцан ярианы загвар, текстийн хураангуй гэх мэт төрөл бүрийн программд ашиглаж байна.

Seq2Seq гэдэг нэр нь үгийн дарааллыг (өгүүлбэр эсвэл эх) оруулаад үгийн гаралтын дарааллыг бий болгодог гэсэн үг юм. Үүнийг recurrent neural network (RNN) ашиглан хийдэг. RNN-ийн энгийн хувилбарыг ховор хэрэглэдэг боловч илүү дэвшилтэт хувилбар болох LSTM эсвэл GRU-ийг ашигладаг. Энэ нь RNN градиент алга болох асуудалтай тулгардагтай холбоотой юм. LSTM-ийг Google-ийн машин орчуулгын системд ашиглагддаг. Энэ нь хугацааны агшинд 2 оролтыг авах замаар үгийн агуулгыг боловсруулдаг. Нэг нь хэрэглэгчээс, нөгөө нь өмнөх гаралтаас ирнэ (гаралт нь оролт болдог).

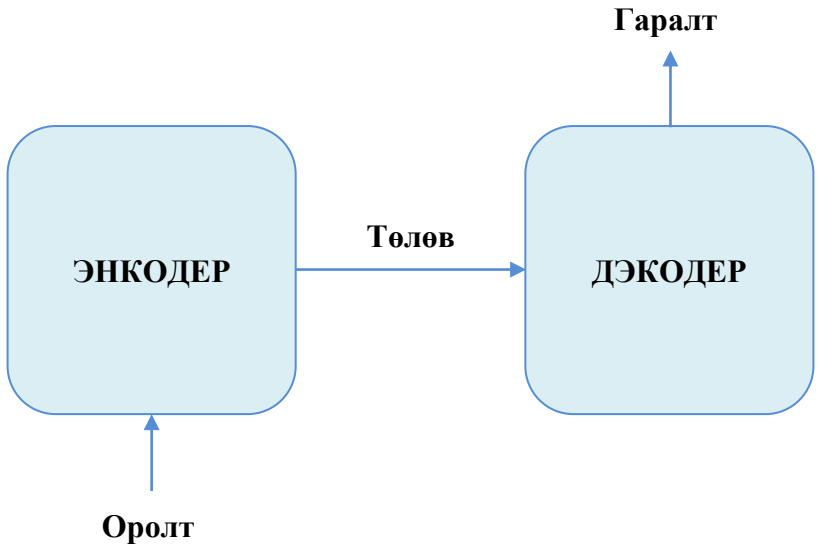
Энэ моделиор бичвэр боловсруулах 2 арга байдаг.

- Тэмдэгтийн боловсруулалт (Character level processing)
- Үгийн боловсруулалт (Word level processing)

Уг модель кодлогч ба код тайлагч гэсэн үндсэн хоёр хэсгээс бүрддэг тул өөрөөр encoder-decoder (энкодер - дэкодер) гэж нэрлэдэг.

Энкодер – Дэкодер (Encoder - Decoder) архитектур

Seq2Seq моделийг үүсгэхэд хамгийн түгээмэл ашиглагддаг архитектур бол энкодер – дэкодер архитектур юм.



Зураг 62. Seq2Seq архитектурын ерөнхий схем

Энкодер (Encoder)

Кодлогч: Энэ нь гүн нейрон сүлжээний давхаргуудыг ашигладаг бөгөөд оролтын үгийг харгалзах далд вектор болгон хөрвүүлдэг. Вектор бүр нь боловсруулж буй үг ба тухайн үгийн агуулгыг илэрхийлнэ.

- Оруулсан дарааллын нэг элементийг хүлээн авч, уг элементийн мэдээллийг цуглуулж, дараагийн алхам руу дамжуулдаг хэд хэдэн давтагддаг нэгжийн стек (LSTM эсвэл GRU эсүүд).
- Асуултанд хариулах системийн хувьд оролтын дараалал нь асуултаас гарч буй бүх үгсийн цуглуулга юм. Үг бүрийг x_i хэлбэрээр илэрхийлэх ба i нь энэ үгийн индекс юм.
- Нуугдмал төлөв h_i-ийг дараах томъёог ашиглан тооцоолно.

$$\boldsymbol{h_t = f(W^{(hh)} h_{t-1} + W^{(hx)} x_t)}$$

Энэхүү энгийн томъёо нь ердийн давтагддаг мэдрэлийн сүлжээний үр дүнг илэрхийлдэг. Дарааллыг боловсруулахдаа өмнөх далд төлөвт тохирсон жин $h\_(t-1)$-г хэрэглэх ба оролтын вектор нь x_t юм.

Энкодер вектор

- Моделийн энкодер хэсгээс гаргаж авсан эцсийн далд төлөв юм. Дээрх томъёог ашиглан энэ векторыг тооцоолно.

- Энэхүү вектор нь декодерыг зөв таамаглал дэвшүүлэхэд туслах үүднээс бүх оролтын элементүүдийн мэдээллийг багтаасан байдаг.
- Энэ нь моделийн тайлагч хэсгийн анхны далд төлвийг илэрхийлдэг.

Декодер (Decoder)

Код тайлагч: Энэ нь кодлогчтой төстэй юм. Энэ нь дараагийн далд векторыг гаргаж, эцэст нь дараагийн үгийг урьдчилан таамаглахын тулд кодлогчийн үүсгэсэн далд векторыг оруулна.

- Хугацааны t агшинд нэг удаа гаралтыг y_t гэж таамаглаж байгаа хэд хэдэн давтагдах нэгжүүдийн стек.
- Дахин давтагдах нэгж бүр өмнөх нэгжээс далд төлөвийг хүлээн авч, өөрийн далд төлөвийг гаргадаг.
- Асуултанд хариулах системд гаралтын дараалал нь хариултаас авсан бүх үгсийн цуглуулга юм. Үг бүрийг y_i байдлаар илэрхийлэх ба i нь үгийн индекс юм.
- Аливаа далд h_i төлөвийг дараах томъёогоор тооцоолно.

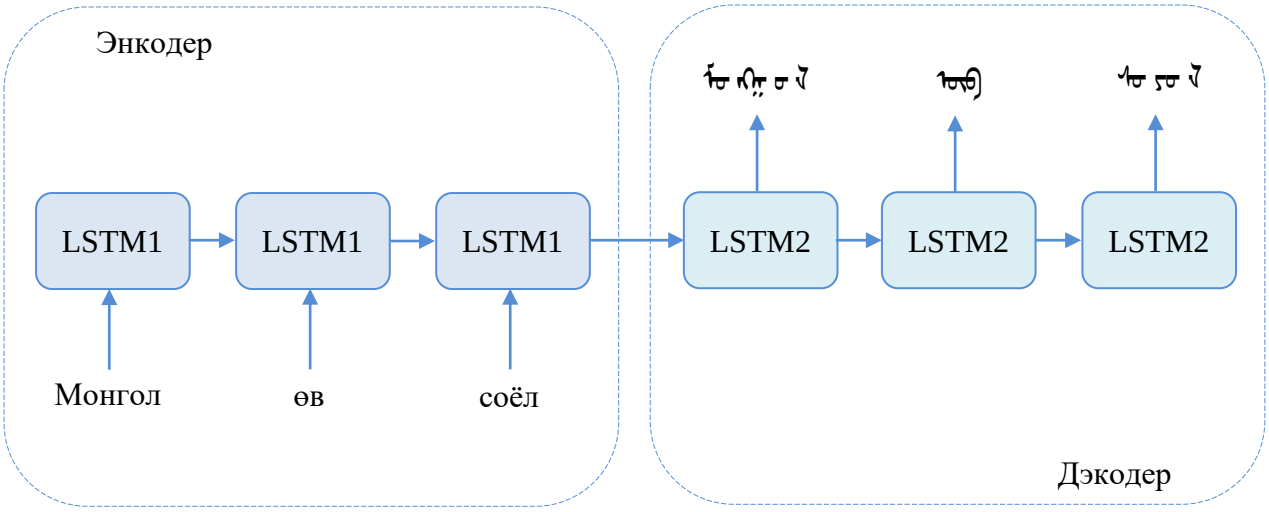
$$\boldsymbol{h_t = f(W^{(hh)} h_{t-1})}$$

- Хугацааны t агшинд гаралтын y_i дарааллыг дараах томъёогоор тооцоолно.

$$\boldsymbol{y_t = softmax(W^S h_t)}$$

Далд төлвийг ашиглан хугацааны тухайн агшинд гаралтыг харгалзах жин (W)-тэй хамт тооцно. Softmax нь эцсийн үр дүнг (жишээ нь асуултанд хариулах системийн үр дүн) тодорхойлоход тусалдаг магадлалын вектор үүсгэдэг.

Энэ моделийн хүч чадал нь өөр өөр урттай дараалал бие биенээ зураглах боломжтой байдагт оршино. Оролт ба гаралт хоорондоо уялдаагүй бөгөөд урт нь ялгаатай байж болно. Энэ шинж нь одоо ийм архитектурын тусламжтайгаар ижил төстэй асуудлуудыг шийдэж болох цоо шинэ арга замыг нээж байна.



Зураг 63. Seq2seq загвар

2 өөр модель үүсгэх псевдо код:

```
emb = Embedding(); lstm = LSTM(); dense = Dense()

input1 = Input(length = Ty)
model1 = Model(input1, dense(lstm(emb(input1))))

input2 = Input(length = 1)
model2 = Model(input2, dense(lstm(emb(input2))))

h = encode model output; x = <SOS>
for t in range(Ty):
      x, h = model2.predict(x, h)
```

LSTM нь к урттай дарааллыг боловсруулж байгаа гэж үзье.

Уг LSTM загвар дарааллыг нэг нэгээр нь уншина. Өөрөөр хэлбэл, бид LSTM загвар к алхам гүйцэтгэж байна гэж ярьдаг.

Дээрх диаграммын хувьд LSTM дараах 3 хэсгээс бүрдэнэ. Үүнд:

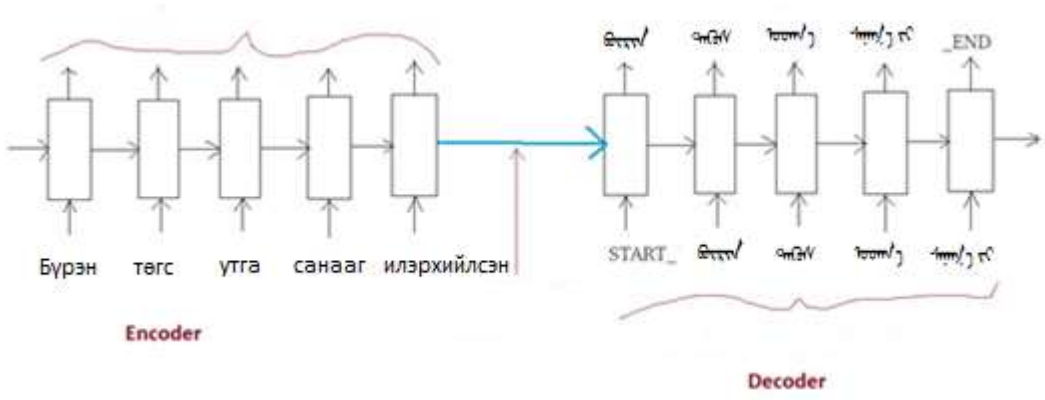
1. Xi => i-р алхам дахь оролтын дараалал
2. h_i болон c_i => LSTM нь алхам бүрт хоёр төлвийг хадгалдаг (h нуугдмал төлөв, c гонхны төлөв). Энэ хоёр нийлж i-р алхам дахь LSTM-ийн дотоод төлвийг илэрхийлнэ.
3. Yi => i-р алхам дахь гаралтын дараалал

Декодер (Decoder) LSTM — Сургалтын горим

Энэ үе шатанд бид декодерыг хэрхэн сургахыг тохируулж өгнө. Оролтын өгүүлбэр “Бүрэн төгс утга санааг илэрхийлсэн” бөгөөд, сургалтын процессын зорилго нь декодерт гаралт бол “[illegible]” өгүүлбэр гэдгийг сургах (заах) явдал юм. Энкодер (Encoder) оролтын дарааллыг үг үгээр уншсантай адилаар декодер (Decoder) гаралтыг үг үгээр үүсгэнэ. Доорх диаграмм.

Гаралтын дараалал => “START_ [illegible] _END”

Сургалтын явцыг бүхэлд (Encoder + Decoder) дараах диаграммаар үзүүлж болно:

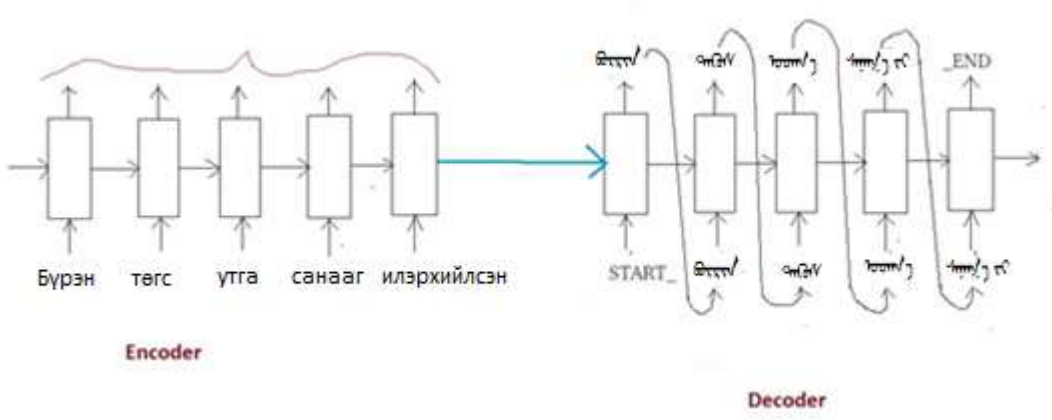


Зураг 64. Seq2Seq сургалтын явц

Хөрвүүлэх алгоритм:

1. Хөрвүүлгийн явцад, тухайн агшинд нэг үгийг хөрвүүлнэ. Тиймээс Decoder LSTM-ийг давталт дотор дуудах бөгөөд бичвэрийг үг тус бүрээр боловсруулна (хөрвүүлнэ).
2. Декодерын анхны төлвийг энкодерын эцсийн төлөвт онооно.
3. Декодерын анхны оролтыг START_ гэж тэмдэглэнэ.
4. Алхам бүрт, декодерийн төлвийг хадгалж авах бөгөөд, дараагийн алхамд зориулан анхны төлвийг оноож өгнө.
5. Алхам бүрт, таамагласан гаралт дараа дараагийн алхмын оролт болно.
6. Декодерт сүүлчийн оролт буюу END_ тэмдэглэгээ орж ирэхэд боловсруулалт хийгээд давталтыг зогсооно.

Хөрвүүлэх процессийг дараах диаграммд үзүүлэв.



Зураг 65. Seq2seq сургалтын загварын дагуу хөрвүүлэх

Кирилл-Монгол бичгийн 90.000 гаруй өгүүлбэрийн зэрэгцээ сан бүрдүүлсэн бөгөөд энэ бүх өгүүлбэрээс сургахад хугацаа их авч байсан тул эхний ээлжинд 2000 зэрэгцээ өгүүлбэрийг ашиглан RNN-ны (learning rate 0.05, 128 нейрон сүлжээ) **seq2seq** загвараар сурган туршив.

Машин сургалт ашиглан хөвүүлэг хийх аргыг ИХ өгөгдөл, БАГА өгөгдөл дээр олон удаа туршиж үзлээ. Үүнээс хамгийн үр дүнтэй гэсэн 4ш туршилтын үр дүнг харуулж байна.

Туршилт 1: 8000 гаруй өгүүлбэр дээр hidden_size=32, dropout_p=0.4, learning_rate=0.05 гэж 3 параметруудыг өөрчлөж туршиж үзсэн. Жишээ болгож 10 өгүүлбэр харууллаа:

> ᠶᠢᠷᠢᠭᠲᠠᠢ ᠡᠲᠭᠡᠡᠳ ᠴ ᠬᠡᠯᠡᠭᠰᠡᠨ ᠪᠠᠢᠨ ᠠ

= яршигтай этгээд
< яршигтай явуулагч <EOS>

> ᠨᠢᠭᠤᠳ ᠠᠯᠪᠠ ᠶᠢᠨ ᠬᠡᠷᠡᠭᠵᠢᠭᠡᠭᠡᠨ ᠵ ᠪᠠ ᠪᠢ

= нууц явуулга хийх
< хийх явуулга хийх <EOS>

> [illegible]

= хөнгөхөн ялалт байгуулах
< гялалзсан ялалт ялалт <EOS>

> [illegible]

= инээдээ барьж ядан
< хэт ядан ядан <EOS>

> [illegible]

= өвдсөн янз үзүүлэх
< яв янз ялгаварлах <EOS>

> [illegible]

= эрүүл мэндийн яам
< гадаад хүчний яам <EOS>

> [illegible]

= амжилттай явуулах
< хэт явуулах <EOS>

> [illegible]

= улаан ягаан
< ягаан ягаан <EOS>

> [illegible]

= зааг ялгаа
< зааг ялгаа <EOS>

> [illegible]

= эвлэрэнгүй маягаар
< янаг амраг <EOS>

Туршилт 2: 8000 гаруй өгүүлбэр дээр hidden_size=64, dropout_p=0.3, learning_rate=0.05 гэж 3 параметруудыг өөрчилж туршиж үзсэн. Жишээ болгож 10 өгүүлбэр харууллаа:

> [illegible]

= ясан хэдрэг
< ясан байх <EOS>

> [illegible]

= санаа зовох юмгүй
< бодох санах юмгүй <EOS>

> ᠬᠦᠯᠢᠭ ᠨᠢ ᠪᠣᠯ ᠰᠡᠢᠷᠢᠭ ᠶᠢᠨ ᠵᠠᠮ ᠢ ᠵᠠᠰᠤᠭᠠᠢ
= улиг болсон зүйлийг яригч
< дэндүү хэлээр яригч яригч <EOS>

> ᠠᠪᠠᠳ ᠴᠠ ᠠᠪᠠᠬᠤ ᠶᠢᠨ ᠪᠠ ᠴᠠ ᠤ
= албадан авах явдал
< албадан авах явдал <EOS>

> ᠬᠥᠳᠡᠯᠮᠥᠷᠢᠶᠢᠨ ᠭᠡᠷ ᠶᠢᠨ ᠠᠵᠢᠯ ᠶᠢᠨ ᠪᠠ ᠴᠢ
= хөдөлмөрийн үйл явц
< хөдөлмөрийн үйл явц <EOS>

> ᠲᠡᠮᠦᠦ ᠭᠡᠬᠦ ᠤ ᠭᠠ ᠪᠤᠷᠤᠭᠤ ᠶᠠᠪᠤᠬᠤ ᠪᠤᠷᠤᠭᠤ ᠶᠠᠪᠤᠬᠤ
= тэмээ гэхээр ямаа гэх буруу явах
< газар буруу эвлэрүүлэх эвлэрүүлэх явах <EOS>

> ᠮᠡᠳᠡᠷᠯᠢᠶᠢᠨ ᠬᠡᠲ ᠶᠠᠳᠠᠷᠭᠠᠠ
= мэдрэлийн хэт ядаргаа
< мэдрэлийн ядаргаа ядаргаа <EOS>

> ᠬᠦᠴᠢᠨᠭᠦᠢ ᠪᠠᠢ ᠬᠡᠪᠴᠢᠶᠡᠨ ᠪᠤᠰ ᠬᠥᠳᠦ ᠰ
= хүйсээр ялгаагүй
< хүйсээр ялгаагүй <EOS>

> ᠠᠪᠠᠷᠭᠠ ᠲᠤᠮ ᠤᠮ ᠤᠮ ᠤ
= аварга том юм
< аар юм юм <EOS>

> ᠥᠪᠡᠷ ᠰ ᠬᠥᠨᠢ ᠬᠤ ᠬᠡᠪᠴᠢᠶᠡᠨ ᠢᠷᠡ ᠭᠠ ᠰ ᠪᠠ ᠵᠠ ᠮᠤ
= өдөр шөнө шиг ялгаатай байх
< өдөр шиг шиг ялгаатай байх <EOS>

Туршилт 3: 90938ш өгүүлбэр дээр hidden_size=128, dropout_p=0.3, learning_rate=0.05 гэж 3 параметруудаа өөрчилж туршиж үзсэн. Жишээ болгож 10 өгүүлбэр харууллаа:

> ᠵᠠ ᠢᠷᠡ ᠵᠠ ᠤ ᠬᠥᠳᠦ ᠰ ᠪᠤ ᠪᠢ
= яаралгүй хийх
< яарахгүй хийх <EOS>

> ᠰᠠᠭᠤ ᠤ ᠬᠡᠰᠡᠭ ᠤ ᠬᠣᠷᠢᠶᠠᠨ ᠳᠡᠭᠡᠭᠰᠢ ᠪᠤᠰ ᠲᠤᠷ ᠳᠤ ᠵᠡᠷᠭᠡ ᠤ ᠬᠣᠯᠪᠣᠭᠤ ᠭᠠᠷᠭᠠᠬᠤ

= шүхрээр онгоцноос яаралгүй үсрэх
< шүхэр онгоцоор явах үсрэх <EOS>

> ᠲᠤ ᠲᠤᠷ ᠳᠤ ᠵᠡᠷᠭᠡ ᠤ ᠬᠥᠳᠦᠭᠡ ᠣᠴᠢᠬᠤ ᠪᠣᠯ ᠬᠣᠯᠤᠭᠠ ᠮᠤᠨ

= яаралгүй дөхөж очих
< яаралгүй дөхөж очих <EOS>

> ᠬᠣᠯᠤᠭᠠ ᠮᠤᠷᠤᠭ ᠬᠤᠷᠤ ᠳᠤ ᠪᠡᠳ ᠪᠤᠯ ᠲᠤ ᠲᠤᠷ ᠳᠤ ᠮᠤᠨ

= урагш тэмүүлэн яарах
< уруу тэмүүлэх яарах <EOS>

> ᠬᠠᠮᠤᠭ ᠲᠤᠷ ᠪᠤᠰ ᠪᠡᠳ ᠨᠢ ᠬᠡᠷᠡᠭ ᠲᠤ ᠲᠤᠷ ᠳᠤ ᠮᠤᠨ

= ухаан зулаггүй яарах
< ухаан ухаан зулаггүй<EOS>

> ᠤ ᠴᠤᠨ ᠲᠤ ᠲᠤᠷ ᠳᠤ ᠮᠤᠨ

= хэт яарах
< хэт яарах <EOS>

> ᠲᠤ ᠣᠨ ᠮᠤᠨ ᠬᠤᠶ ᠲᠤᠷ ᠪᠤᠯ ᠲᠤ ᠲᠤᠷ ᠳᠤ ᠮᠤᠨ

= явахдаа яарах
< явахдаа яарах <EOS>

> ᠠᠪᠠᠰ ᠴᠤᠨ ᠲᠤ ᠬᠤ ᠲᠤ ᠲᠤᠷ ᠳᠤ ᠮᠤᠨ

= асар их яарах
< асар их их <EOS>

> ᠲᠤ ᠲᠤᠷ ᠴᠠᠭ ᠨᠢ ᠪᠠᠷᠢᠭ ᠳᠤᠭᠤᠷ ᠲᠤᠷ ᠪᠤᠯ ᠬᠤᠭᠤ ᠲᠤ ᠤ

= яармаг худалдааны газар
< яармаг худалдаа газар <EOS>

> ᠬᠥᠷᠭᠡᠨ ᠪᠤ ᠲᠤ ᠤᠭᠡᠯ ᠤ ᠪᠤᠯ ᠪᠠ ᠲᠤ ᠲᠤᠷ ᠪᠤᠯ ᠲᠤ ᠲᠤᠷ ᠴᠠᠭ ᠨᠢ

= өргөн хэрэглээний барааны яармаг
< өргөн хэрэглээ барааны яармаг <EOS>

Туршилт 4: 90938ш өгүүлбэр дээр hidden_size=64, dropout_p=0.4, learning_rate=0.05 гэж 3 параметруудаа өөрчилж туршиж үзсэн. Жишээ болгож 10 өгүүлбэр харууллаа:

> ᠲᠤ ᠲᠤᠷ ᠳᠤ ᠵᠡᠷᠭᠡ ᠤ ᠤᠨ ᠺ

= яаралгүй хийх
< хийх хийх <EOS>

> ᠰᠤᠭᠤᠵᠠᠷ ᠤᠨ ᠬᠡᠯᠡᠨᠡᠬᠡᠴᠡ ᠭᠠᠵᠠᠷ ᠲᠤᠰ ᠲᠤᠰ ᠤᠨ ᠬᠡᠯᠡᠨᠡᠬᠡᠴᠡ ᠵᠠᠯᠭᠠᠬᠤ
= шүхрээр онгоцноос яаралгүй үсрэх
< шүхэр онгоцоор явах явах<EOS>

> ᠲᠤ ᠲᠡᠮᠳᠡᠭ ᠤ ᠬᠥᠳᠥᠭᠡ ᠪᠤᠢ ᠬᠥᠳᠥᠯᠬᠥ ᠨᠢ
= яаралгүй дөхөж очих
< яаралгүй очих очих<EOS>

> ᠬᠥᠳᠥ ᠮᠤᠷᠤᠨ ᠳᠤ ᠪᠠᠭᠤ ᠪᠠᠭᠤ ᠠᠷ ᠲᠤ ᠲᠡᠮᠳᠡᠭ ᠲᠤ ᠮᠤᠭ
= урагш тэмүүлэн яарах
< уруу дээр яарах <EOS>

> ᠬᠤᠭᠤᠴᠠᠭ ᠲᠡᠮᠳᠡᠭ ᠦᠭᠡᠢ ᠬᠥᠳᠥᠯᠬᠥ ᠤ ᠲᠤ ᠲᠡᠮᠳᠡᠭ ᠲᠤ ᠮᠤᠭ
= ухаан зулаггүй яарах
< ухаан ухаан зулаггүй<EOS>

> ᠤ ᠴᠠᠭ ᠲᠤ ᠲᠡᠮᠳᠡᠭ ᠲᠤ ᠮᠤᠭ
= хэт яарах
< хэн явах <EOS>

> ᠲᠤ ᠪᠠ ᠮᠤᠭ ᠠᠪ ᠲᠡᠮᠳᠡᠭ ᠠᠷ ᠲᠤ ᠲᠡᠮᠳᠡᠭ ᠲᠤ ᠮᠤᠭ
= явахдаа яарах
< наран явах<EOS>

> ᠠᠰᠠᠭᠠᠭ ᠤᠨ ᠬᠤ ᠲᠤ ᠲᠡᠮᠳᠡᠭ ᠲᠤ ᠮᠤᠭ
= асар их яарах
< асар их их <EOS>

> ᠲᠤ ᠲᠡᠮᠳᠡᠭ ᠬᠥᠳᠡᠭ ᠦᠭᠡᠢ ᠪᠤᠯᠤᠭᠠᠨ ᠲᠡᠮᠳᠡᠭ ᠲᠡᠢ ᠭᠠᠵᠠᠷ ᠤ
= яармаг худалдааны газар
< яармаг газар газар <EOS>

> ᠬᠥᠬᠡᠨ ᠠᠢ ᠤᠨ ᠲᠤᠪᠠᠨ ᠤ ᠠᠢᠮᠠᠭ ᠪᠤᠷ ᠲᠤ ᠲᠡᠮᠳᠡᠭ ᠲᠡᠢ ᠲᠤ ᠲᠡᠮᠳᠡᠭ ᠬᠥᠳᠡᠭ ᠦᠭᠡᠢ
= өргөн хэрэглээний барааны яармаг
< өргөн хэрэглээ байгаа яармаг <EOS>

4.3 Бүлгийн дүгнэлт

Кирилл болон монгол бичгийн хооронд харилцан хөрвүүлэх алгоритмыг монгол хэлний дүрэмд, статик өгөгдөлд суурилсан болон машин сургалтын аргыг ашиглан тус тус боловсруулан туршлаа. Туршилтын үр дүнд үндэслэн дараах дүгнэлтийг хийлээ.

1. Залгамал бүтэцтэй монгол хэлний үгийн бүтцийг нь үндэс+нөхцөл1+ нөхцөл2+ ... + нөхцөлN гэж үзэж болно. Энхүү үгийн бүтцийн онцлогоос шалтгаалан кирилл, монгол бичгийн хооронд хөрвүүлэхдээ үгийг бүтцээр нь задлах мөн залгах үйлдэл хийнэ.
2. Энэхүү хоёр бичиг хооронд хөрвүүлэхдээ нэг бичгээр өгөгдсөн үгийг бүтцээр нь задалж, үгийн үндэс болон нөхцөлүүдийг олно. Ингээд өгөгдлийн сангаа ашиглаж үгийн үндэс ба нөхцөлүүдийн нөгөө бичгээрх хэлбэрийг тодорхойлно. Эцэст нь олж тодорхойлсон үндсийг өгөгдсөн нөхцөлүүдээр хувилгах замаар нөгөө бичигтээ хөрвүүлнэ. Кирилл бичгээс монгол бичигт хөрвүүлэх үед кирилл үгийн бүтэц нь олон хэлбэрээр задрах болон үгийн утгаас хамаарч монгол бичгээр ялгаатай бичигдэх боломжтой байдаг. Иймээс үгийн утга таних асуудал чухлаар тавигдана. Монгол бичгээс кирилл бичгээс хөрвүүлэх үед монгол бичгээр бичигдсэн оноосон нэрийг ялгаж таних, мөн үгийн утга таних асуудал чухлаар тавигдаж байна.
3. Туршилт хийсэн бичвэрийн зөв бичих дүрмийн алдаанаас болж хөрвүүлээгүй буюу буруу хөрвүүлсөн үг байгаа учраас хөрвүүлэхийн өмнө хоёр бичгийн бичвэрийн алдааг шалган зөв бичих дүрмийн дагуу нэг бүрчлэн засаж дахин хөрвүүлэлт хийх нь зүйтэй.
4. Монгол хэлний хоёр бичгийн бичвэрийн хөрвүүлгийн хувьд үүсмэл хэлэнд голчлон хэрэглэгддэг дүрэмд суурилсан аргыг хэрэглэхэд илүү оновчтой болох нь бидний судалгааны ажлаас харагдсан бөгөөд кирилл болон монгол бичиг нь өгүүлбэр зүйн бүтцийн хувьд адилхан тул монгол хэлний эдгээр хоёр бичгийн хувьд үг зүйг нь загварчлахад хангалттай байсан.
5. Бичвэр боловсруулах ажлын үр дүн нь тэмдэгтийн кодлолтоос (латин, кирилл, монгол бичиг гэх мэт) хамааралгүй бөгөөд гол нь үгийн сангийн файлаа хэр хангалттай бүрдүүлж, зөв зүйтэй ангилав, дүрмээ хэр зөв тодорхойлж загварчлав гэдгээс шууд хамаарч байна. Тиймээс монгол хэлийг кирилл үсэг, уламжлалт монгол бичиг, латин үсгийн алинаар нь кодолж, түлхүүрдэж байгаагаас үл хамааран бидний боловсруулан үг зүйн загварчлалын хэрэгслийг ашиглаж болж байна.
6. Кирилл болон монгол бичгийн хооронд харилцан хөрвүүлэх алгоритм боловсруулан амжилттай туршлаа. Мөн төгсгөлөг төлөвт хувиргагч ашиглах аргачлал боловсруулан хэрэгжүүллээ. Үгийн үндэс нь тухайн үгийн үндсэн утгыг хадгалах тул сургалтын жишээнээс үгийн үндсийг ялган хадгалах нь зүйтэй гэж үзлээ. Мөн кирилл бичигт ижил бичлэгтэй нэр ба үйл үг байх боломжтой тул сургалтын жишээнд нэр ба үйл үгийг заах нь илүү үр дүнтэй байна.
7. Үг зүйн шинжилгээ хийхээр сонгож авсан арга үр дүнтэй болсон бөгөөд Монгол хэлэнд тохирох нь батлагдлаа. Мөн байгуулсан өгөгдлийн сан нь монгол хэл шинжлэлийн чиглэлээр хийх ажлын суурь болж чадахуйц зөв бүтэцтэй болсон байна.

8. Их хэмжээний өгөгдөл дээр дараалалтай холбоотой асуудлыг шийдэхэд RNN (Recurrent Neural Network) нь хамгийн тохиромжтой машин сургалтын арга бөгөөд яриа таних, эх хэлний боловсруулалт (NLP), хугацаанаас хамаарсан таамаглал дэвшүүлэх зэрэг олон төрлийн хэрэглээтэй. RNN архитектурын тусгай арга болох Sequence to Sequence (seq2seq) загварыг машин орчуулга, асуултанд хариулах, чатбот үүсгэх, бичвэрийг хураангуй болгох гэх мэт хэлний нарийн төвөгтэй асуудлыг шийдвэрлэхэд ихэвчлэн ашигладаг. Бид ч гэсэн энэ аргыг ашиглан монгол хэлний хоёр бичгийн бичвэр хооронд хөрвүүлэх туршилтыг хийсэн. Ингэхдээ I бүлэгт дурдсан кирилл-монгол бичгийн 90.000 гаруй холбоо үгийн болон харгалзсан 5000 өгүүлбэр бүхий сангаа ашигласан бөгөөд энэ сангаас сургахад хугацаа их авч байсан тул эхний ээлжинд 2000 холбоо үг, өгүүлбэрийг ашиглан RNN-ны (learning rate 0.05, 128 нейрон сүлжээ) seq2seq загвараар сурган туршив. Ингэхдээ машин сургалт ашиглан хөвүүлэг хийх аргыг ИХ өгөгдөл, БАГА өгөгдөл дээр олон удаа туршиж хамгийн үр дүнтэй гэсэн туршилтын үр дүнг тайланд тусган харуулсан.

НОМ ЗҮЙ

[1] МУБИС, Монгол Судлалын Сургууль, Монгол Хэлшинжлэлийн Тэнхим, “Орчин цагийн монгол хэл”, Улаанбаатар хот, МУБИС, Монгол судлалын сургууль, 2004.

[2] U. Batbayar, Ch. Lodoiravsal and R. Amartuvshin, “Recognition of printed Traditional Mongolian script”, in Conference proceeding MITA 2011, Ulaanbaatar, Mongolia, 2011.

[3] Д. Бямбадорж, “Монгол хэлний хэлбэр судлал”, Улаанбаатар хот, УБИС-ийн хэвлэх үйлдвэр, 2006.

[4] Б. Батзолбоо, Ю. Намсрай ба Ш. Чоймаа, “Текстийг кирилл, монгол бичгийн хооронд хөрвүүлэх системийн үгийн сан”, ШУТИС-ийн эрдэм шинжилгээний бүтээлийн эмхэтгэл 9/89, Улаанбаатар хот, 2006.

[5] Ш. Чоймаа, М. Баярсайхан, Э. Мөнх-Учрал ба С. Батхишиг, “Монгол хэлний хэл зүйн толь бичиг”, Улаанбаатар хот, МУИС-ийн хэвлэх үйлдвэр, 2006.

[6] Э. Мөнх-Учрал, “Хөрвүүлэх програмд зориулсан монгол хэлний судалгаа”, Улаанбаатар хот, МУИС, Монгол Хэл Соёлын Сургууль, докторын зэрэг горилсон диссертаци, 2010.

[7] J. Purev, Z. Tsolmon, Ch. Altangerel and Cheol-Young, “PC-KIMMO based description of Mongolian morphology”, Ulaanbaatar, 2005.

[8] Ch. Altangerel and B. Adiyatseren, “Two level rules for Mongolian Language” in Conference proceedings MITA 2011, Ulaanbaatar, 2011.

[9] R. Wicentowski, “Modelling and Learning Multilingual Inflectional Morphology in a Minimally Supervised Framework”, Baltimore, Maryland: The Johns Hopkins University, a thesis for doctor, 2002.

[10] D. Jurafsky and J.H. Martin, “Speech and Language Processing”, New Jersey, Pearson Education, Inc., 2009.

[11] J.G. Edward Barton, “The computational complexity of two-level morphology” in Massachusetts institute of technology artificial intelligence laboratory, 1985.

[12] K. Koskenniemi, “Two-level morphology: a general computational model for word-form recognition and production”, Helsinki, Finland: University of Helsinki, a thesis for doctor, 1983.

[13] H. Trost, “Computational morphology,” 2009. http://ccl.pku.edu.cn/doubtfire/NLP/Lexical_Analysis/Word_Lemmatization/Introduction/Computational%20Morphology.htm, 2013.

[14] J. Hans, “A two-level engine for tagalog morphology and a structured XML output for PC-KIMMO”, Brigham Young University. Department of Linguistics and English Language, 2004.

[15] Kasetsart University, Bangkok, Thailand, “Computational morphology,” 2006. http://naist.cpe.ku.ac.th/LAICS-NLP/document/170ct06/rooml/2.1.1Computational Morphology%28BaliRanaivo%29.pdf, 2013.

[16] Б. Отгонбаяр, “Монгол хэлний ярианы синтезийн параметрүүдийг тогтоох судалгаа”, Улаанбаатар хот, Докторын диссертаци, 1996.

[17] О. Бат-Энх, “Ярианы синтезийн шинэ арга боловсруулах”, Улаанбаатар хот, Докторын диссертаци, 2001.

[18] K. Y. Mi, “The Model of Korean-Mongolian Machine Translation”, Ulaanbaatar, Doctoral thesis, 2012.

[19] D. Uuganbaatar, “Research on Cyrillic and Mongolian script's morphology and conversion system”, Khuhhot, Doctoral thesis, 2014.