

# Монгол хэлний Кирилжин бичгийн үг хувиллын алгоритм ба хэрэгжүүлэлт

Э.Авирмэд<sup>1</sup>, Ш.Нямбаа<sup>2</sup>

МУИС-ийн Математик Компьютерийн Сургууль  
Програм хангамжийн тэнхим  
avirmed@smcs.num.edu.mn<sup>1</sup>, nyambaa@num.edu.mn<sup>2</sup>

*Хураангуй*—Компьютерт эх хэлээрээ их хэмжээний мэдээлэл, баримт бичиг боловсруулж бэлтгэхээс гадна интернэт ашиглан хүссэн мэдээллээ олж унших, и-мэйл бичих, олон нийтийн сүлжээгээр харилцах зэрэг олон боломжууд байдаг.

Энэ хөгжлийн хажуугаар гарж буй нэг асуудал бол монгол хэлний бичгийн соёл алдагдаж зөв бичих дүрмийн алдаа гарах болсон. Хүмүүс ихэнхдээ санаандгүй болон дүрмийн мэдлэг дутмагаас болж үгийг алдаатай бичдэг. Зарим ном, сонин сэтгүүлд ч зөв бичих дүрмийн алдаа ажиглагддаг. Энэ асуудлыг шийдэхийн тулд кирилжин монгол бичгийн алдааг шалган засварлах програмын хэрэгцээ гарч ирж байна.

Бидний ажил Академич Ц.Дамдинсүрэнгийн монголын оюуны соёлд оруулсан нэгэн томоохон хувь нэмэр болсон кирилжин монгол бичгийн зөв бичихүйн дүрмийг үндэслэн мөр төрлийн өгөгдөл болох ямар нэг үгээс хамгийн урт дэд дарааллыг олох аргыг ашиглан кирилжин бичгийн үг хувиллын алгоритмыг боловсруулж хэрэгжүүлсэн тухай өгүүлнэ.

*Түлхүүр үг*—Эх хэлний боловсруулалт; зөв бичих дүрмийн шалгуур; нийлэг алгоритм; хоёртын төгсгөлөг мод; хамгийн урт дэд дараалал;

## I. УДИРТГАЛ

Эх хэлний боловсруулалт (ЭХБ) [NLP -Natural language processing] нь хүний хэл болон компьютер хоорондын харилцан үйлчлэл, эх хэлний өгөгдлүүд дээр математик, статистик боловсруулалт хийж судалдаг компьютерийн шинжлэх ухаан, хиймэл оюун ухааны нэг салбар ухаан юм.

Энэ чиглэлийн судалгааны ажил эрт үеэс эхлэлтэй ч үндсэндээ 1950 оноос ЭХБ-н түүх эхэлсэн гэж үздэг. 1950 онд Алан Туринг өөрийн алдарт “Тооцоолох Машин ба Оюун ухаан” гэсэн бүтээлээ хэвлүүлсэн. ЭХБ-д тухайн хэлний зөв бичих дүрмийн алдаа шалгадаг модуль маш их үүрэгтэй.

Дэлхийн ихэнх улс орон өөрийн хэлний зөв бичгийн алдааг засах програмыг тодорхой түвшинд хөгжүүлсэн. Тэдгээрээс нийтэд хамгийн өргөн түгсэн нь англи хэлний алдаа засагч юм. Энэ нь бидний өдөр тутам хэрэглэдэг хэрэглээний програмуудад өргөн ажиглагддаг.

Энэ чиглэлийн судалгааны ажил МУИС болон ШУТИС-ийн салбар сургуулиуд дээр бакалавр, магистр, докторын

ажлаар гүйцэтгэсэн олон ажлууд[4] хийгдсэн ч өнөөг хүртэл бүх нийтийн хэрэглэнд ашиглаж байгаа програм цөөхөн байна.

Аливаа нэг хэлийг компьютерээр боловсруулах процесст хүний яриа таних (Speech Recognition), тухайлбал яриаг текст болгох STT (speech to text), текстийг яриа болгох TTS (text to speech) болон оптик тэмдэгт таних OCR (Optical Character Recognition) технологиор бичвэрийг зурагнаас ялгах зэрэг асуудалд тухайн хэлний үгийг зөв бичих дүрмийн шалгуур зайлшгүй хэрэгцээтэй болдог.

Ийнхүү монгол хэлийг компьютерээр боловсруулахын тулд зайлшгүй үгийг зөв бичих дүрмийн шалгуур (spelling check) шаардлагатай.

Монгол хэлний зөв бичгийн дүрмийн шалгуур програмыг хэл зүйн болон хайлт оптимизацийн оновчтой алгоритм гаргаж хэрэгжүүлэх, улмаар бүтээгдэхүүн болгоход ажлын агуулга оршино.

## II. АСУУДЛЫН ШИЙДЭЛ

Кирилжин бичгийг компьютерээр боловсруулахад кирилжин бичгээр бичих нийлэг алгоритм (to-Cyrillic composing), кирилжин бичгээс хөрвүүлэх задлаг алгоритм (from-Cyrillic decomposing) гэсэн үндсэн хоёр төрлийн алгоритм байх бөгөөд аль аль нь уг бичгийн зөв бичихүйн тогтолцоон дээр суурилсан байдаг. Эхнийх нь уг тогтолцоог шууд хэлбэрээр тусгадаг бол хоёрдахь нь урвуу хэлбэрээр тусгадаг. Үсгийн дүрмийн тогтолцоог загварчлан харуулахын тухайд нийлэг алгоритм нь илүү тохиромжтой[1].

### 2.1 КИРИЛЖИН БИЧГЭЭР ЗӨВ БИЧИХ НИЙЛЭГ АЛГОРИТМ, ҮГИЙН САН

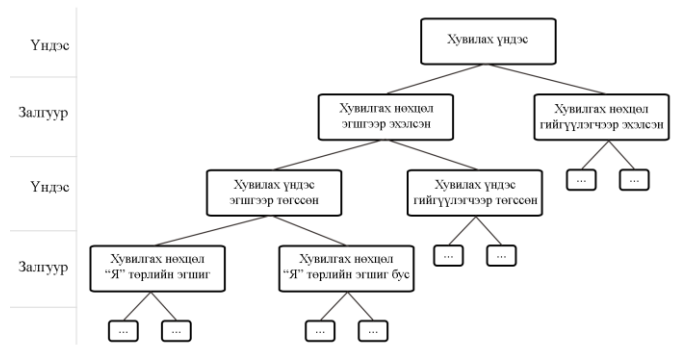
**2.1.1 Кирилжин бичгээр бичих нийлэг алгоритмын хоёртын мод дүрслэл:** Кирилжин бичгийг 1946 онд Монголын албан ёсны бичиг болгосон. Кирилжин бичигт үг хувиллын 66, үсгийн 189 үндсэн дүрэмтэй. Үүн дээр гажилтын дүрмийг нэмбэл нийтдээ 270 орчим дүрмээс бүтдэг. Харин мянга орчим жилийн өмнөөс уламжлагдан ирсэн монгол бичгийн зөв бичих зүй нь эгшиг зохицох ёс, дэвсгэрлэх ёс, холбох эгшигийн гэсэн үндсэн гурван дүрэмтэй байдгаараа бусад бичгээс ялгарах давуу талтай.

Кирилжин бичвэрт хоёр талаасаа зайгаар тусгаарлагдсан хэсэг болох үгийн хэлбэрийг хувилсан хэлбэр (inflected form - IF), эс хувилсан хэлбэр (uninflected form - UF) гэж хоёр хувааж болох бөгөөд хувилсан хэлбэр нь хувилах үндэс (handling stem - HS) хийгээд хувилгах нөхцөл (processing infix - PI) гэсэн хоёр үндсэн бүрдүүлэгчийн нийлмэл цогц юм. Үүнийг томъёолон үзүүлбэл:

$$HS + PI(+ PI...) \rightarrow IF \quad (1)$$

Энэ хоёр бүрдүүлэгчийг тус бүрийнх нь онцлогоос хамаарч өөр өөрөөр тодорхойлох шаардлагатай. Тухайлбал, хувилах үндсийг түүний төгсгөлөөр нь, харин хувилгах нөхцлийг түүний эхлэлээр нь тодорхойлно[1].

Дээрх (1) томъёог өгөгдлийн бүтцийн төгсгөлөг хоёртын мод (ТХМ) хэлбэрээр дүрслэж болно. Энэ дүрслэлийн агуулга нь монгол хэлний үг зүйн хувилалд үгийн төгсгөл болон залгах нөхцөл нь эгшиг буюу гийгүүлэгчийн хоёрын аль нэг нь байдаг учир хоёртын мод төрлөөр, HS болон PI илэрхийлэлүүдийг холбох дүрэм, шаардлагууд нь төгсгөлөг байгаад оршино. Үг хувиллийн ТХМ-ны дүрслэлийн хураангуй хэлбэрийг зураг 1-д үзүүлэв[1].



Зураг 1. Төгсгөлөг хоёртын модны эхлэлийн хэсэг

**2.1.2 Монгол хэлний үгийг хувилгах нөхцөлөөр бүтээх:**

Монгол хэлний үндсэн нэгж болох өгүүлбэрийн бүтэц дэх үгсийн сан, утгазүй, хэлзүйн онцлог шинжээр нь бүлэглэн ангилсаныг үгсийн аймаг гэдэг. Үгсийн аймгийг мэдсэнээр үгийн санг зохион байгуулах, дагавар бүтээврээр зөв ангилж өгөхөд хэрэгтэй.

Монгол хэлний үгийн бүтэц ерөнхий нэг дараалалтай байдаг. Тиймээс нэр үг болон үйл үгийн дагавар нөхцөлүүд дурын байдлаар сэлгэж үг бүтээдэггүй.

Хувилгах нөхцөлүүдийг нэрийн нөхцөл, үйлийн нөхцөл гэж хоёр хэсэг ялгасан. Түүнчлэн нэрийн нөхцөл залгах үед үгийн хэрэглэгдэх шинжээс хамаарч нууц 'н', 'г', 'х' ил гарах тохиолдлууд гардаг.

Үгийн нөхцөлөөр хувилахад үндсэндээ дараах хоёр бүтцийг ойлгож болно. Үүнд:

- *Нэрийн нөхцөлөөр хувилах үгийн бүтэц :*

Үндэс + [дагавар] + [олон тооны нөхцөл] + [тийн ялгалын нөхцөл] + [хамаатуулах нөхцөл]

- *Үйлийн нөхцөлөөр хувилах үгийн бүтэц :*

Үндэс + [дагавар] + [хэвийн нөхцөл] + [байдлын нөхцөл] + [холбох болон төгсгөх нөхцөл]

Монгол хэлний дагавар бүтээврийн зөв бичих аргачлалыг хувилах үндэс буюу язгуурын төгсгөл, хувилгах нөхцөл буюу бүтээврийн эхлэлийн шинжээр авч үзэн тохирох дүрмийн дагуу залгана.

**2.1.3 Үгийн сангийн дүрслэл:** Кирилжин бичгийн алдаа засагч нь үгийн үндсийг нөхцлүүдээр хувилгадаг тул үгийн санд язгуур болон үндэс хэлбэрээр хадгална. Дагавраар хувилсан үгийг үндэс хэмээн үзэж үгийн санд шууд оруулна[5]. Жишээ: *явал, барилга, чихэвч, жаргалан* г.м.

Үгийг санд хадгалахдаа тухайн үгэнд залгагдаж болох нөхцөлийн төрөл, оноосон нэр болон нууц 'н', 'г' авах эсэхийг тодорхойлсон атрибутыг зааж өгнө. Тухайлбал, эхний атрибут нь нэрийн эсвэл үйлийн нөхцөлийн аль төрлөөр хувилахыг, дараагийн атрибут нь балархай эгшиг гээгдэхгүй оноосон нэр заана, харин гурав дахь атрибут нь дүрмийн биш хувилалтай нууц 'н', 'г' авах эсэхийг тодорхойлно.

*Нэрийн нөхцөл авах үг:* **Noun : N<sub>1</sub> : N<sub>2</sub> : N<sub>3</sub>**

Энд: - N<sub>1</sub>-нь нэр үг дурын олон тооны хувиллаар хувирдаггүй учир олон тооны нөхцөлийг ялгах агуулгатай. Жишээлбэл: N<sub>1</sub>=1 үед -ууд<sup>2</sup>, N<sub>1</sub>=2 үед -чууд<sup>2</sup>, N<sub>1</sub>=3 үед -с. Харин N<sub>1</sub>=0 бол олон тооны нөхцөл авахгүй. Учир нь -д, -нар гэсэн олон тооны нөхцөл авдаг үгийн хувилсан хэлбэрийг үгийн санд шууд оруулахаас гадна олон тооны нөхцөл авдаггүй үгс байдаг. N<sub>1</sub>=-1 тохиолдолд нөхцөл авдаггүй дайвар үгийг заана.

- N<sub>2</sub>-атирибут нь оноосон нэр бол N<sub>2</sub>=1 байна.
- N<sub>3</sub>-атрибут нууц 'н', 'г' шаардах эсэхийг тодорхойлно. N<sub>3</sub>=0 үед нууц 'н', 'г'-гүй, N<sub>3</sub>=1 үед 'н', N<sub>3</sub>=2 бол 'г' илрэхийг заана.

Жишээ нь: *дэвтэр:1:0:1, нөхөд: 0:0:0, нь :-1:0:0, минь :-1:0:0, Улаанбаатар :2:1:0, сүрьеэ :1:0:2*

*Үйлийн нөхцөл авах үг:* **Verb : V<sub>1</sub>**

Энд: - V<sub>1</sub>-нь бусдаар үйлдүүлэх хэвийн нөхцөлийг тэмдэглэнэ. Үйл үг нь олон тоотой адил бусдаар үйлдүүлэх хэвийн нөхцөлийн ялгаатай хувилбаруудаар хувилдаг. V<sub>1</sub>=4 үед -уул<sup>2</sup>, V<sub>1</sub>=5 үед -лга, V<sub>1</sub>=6 үед -га, V<sub>1</sub>=7 үед -аа

Жишээ нь: *хар:4, таа:5, гар:6, хат:7*

Түүнчлэн хоёр болон түүнээс дээш утга илэрхийлдэг үгийг үгийн санд атрибутаар нь ялгаж дахин оруулсан.

Жишээ нь: *хар:4 (харах үйл үг), хар:1:0:0 (хар өнгө), сур:6 (сурах үйл үг), сур:1:0:0 (сур харваа)*

2.2 АЛДААТАЙ ҮГИЙГ ХАЙЛТ

Хэрэглэгчийн бичсэн үгийг алдаатай эсэхийг тогтоохын тулд үгийн сангийн бүх үгтэй жиших шаардлага гардаг. Энэ жишилтийн хамгийн бэрхшээлтэй асуудал нь тухайн үг хэд хэдэн төрлийн алдаатай бичигдсэн хэлбэрээс ойролцоо язгуурыг олох асуудал юм.

Бичвэрт гардаг түгээмэл алдаа:

- Үсэг дутуу бичих
- Үсэг илүү бичих
- Үсгийн байрыг солих
- Дүрмийн алдаа гаргах

Эдгээр алдаа орсон бичвэрээс үгийн ойролцоо язгуур үндсийг олох нь нилээд бэрхшээлтэй. Энэ асуудлыг шийдвэрлэхэд *Хамгийн урт дэд дарааллыг олох* (Longest common subsequence) арга тохиромжтой гэж үзлээ[5].

**2.2.1 Хамгийн урт дэд дарааллыг олох арга:** Хамгийн Урт Дэд Дарааллыг (ХУДД) олох арга нь *хоорондоо ялгаатай хоёр мөр төрлийг (string) харьцуулж хамгийн урт дэд дараалал буюу мөр төрлийг олдог*[2,3]. Энэ арга нь ХУДД-ыг олохдоо рекурсив алгоритмаар хялбар шийдэж болох боловч энэ нь өөрөө өөрийгөө дуудаж ажилдаг учир хэд хэдэн асуудал үүсдэг. Тухайлбал, харьцуулах гэж байгаа хоёр мөр төрлийн хувьд хоёуланд нь орсон тэмдэгтээс ороогүй тэмдэгтүүд их байвал олон рекурсив ажиллаж хугацаа алдах тул динамик програмчлалын нилээд сонирхолтой асуудал юм[3]. ХУДД-г олох аргыг хэрэгжүүлэгч техникийг цаашид ХУДД-ын алгоритм гэж нэрлэе.

**2.2.2 Дэд дарааллыг шалгах:** ХУДД-г олох алгоритмыг тодорхойлохоос өмнө энгийн жишээ үзье. Өгөгдсөн богино болон урт текстүүдийг харьцуулна. Үүний тулд богино текстэнд байгаа үсгүүдийн дараалал хэвээр (дунд нь өөр тэмдэгтүүд орсон байж болно) урт текстэнд агуулагдаж байгаа эсэхийг шалгах шаардлагатай. Хэрэв агуулагдаж байвал богино текстийг урт текстийн дэд дараалал гэнэ. Жишээ нь “од” гэсэн үгийг “отгон дүү”-тэй харьцуулж үзэхэд “**Отгон Дүү**” томоор бичигдсэн үсэгнүүдийн байрлал зөв дараалсан учир урт текстийн дэд дараалал болно.

Ерөнхий тохиолдолд богино текстийн тэмдэгт бүрээр дарааллуулан хайлт хийнэ. Энд буцах хайлт хийх шаардлагагүй. ХУДД-г олох алгоритмын псевдо-код дараах хэлбэртэй:

```
subseq (char *P, char *T) {
    while(*T != '\0')
        if(*P == *T++ && *++P == '\0')
            return TRUE;
    return FALSE;
}
```

Алгоритм 1. Дэд дараалал олох

**2.2.3 Рекурсив хайлтын арга:** ХУДД-ыг олоход динамик програмчлалын рекурсив арга нь нилээд зөв шийдэл болдог. Динамик програмчлалын гол санаа нь *хэрхэн хайх*

*нь чухал биш, үр дүнтэй шийдлийг олох замыг бидэнд өгөхөд оршдог.*

ХУДД олох алгоритмыг “*сургууль*”, “*сургааль*” жишээн дээр байгуулъя:

```
с у р г у у л ь
| | | |   | |
с у р г а а л ь
```

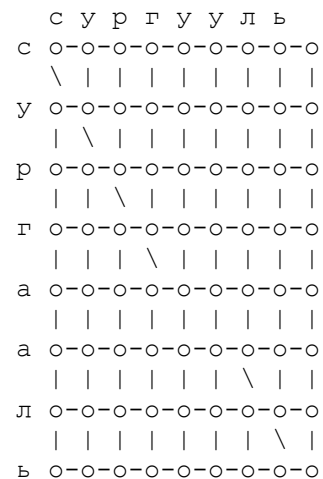
Эхний үгэнд байгаа үсгүүдийг хоёрдахь үгийн тохирох үсгүүдтэй шулуунаар холбох ба шулуунууд огтлолцож болохгүй. Холболтын дараа үүссэн дараалал нь дэд дараалал болно. Энд шулуунууд огтлолцохгүйгээр хэд хэдэн төрлөөр үсгүүдийг холбож дэд дарааллуудыг үүсгэх боломжтой. Энэ техникийн псевдо-кодыг алгоритм 2-т үзүүлээ.

```
int lcs_length(char *A, char *B){
    if(*A == '\0' || *B == '\0')return 0;
    else if(*A == *B) return 1 +
        lcs_length(A+1, B+1);
    else return max(lcs_length(A+1, B),
        lcs_length(A, B+1));
}
```

Алгоритм 2. ХУДД олох рекурс арга

Энэ нь хэдий зөв шийдэл боловч хугацаа их шаарддаг. Жишээлбэл хоёр текстэнд нэг ч ижил үсэг байхгүй бол гуравдахь мөр олон дахин ажиллах болно. Хугацаа хоёр текстийн урт болох m, n-ээс хамаарах ба m=n байвал ойролцоогоор O(2<sup>n</sup>) хугацаанд ажиллана.

**2.2.4 Графын холбоост зам:** А ба В үгсийн үсгүүдийг орой гэж үзэн ижил үсгүүдийг холбож L[m][n] гэсэн хоёр хэмжээст хүснэгт байгуулна.



Дээрх графд дэд дараалал нь диагоналийн дагуу үүсч байгаа нь харагдана. Өөрөөр хэлбэл хэвтээ болон босоо ирмэгүүдийн хувьд 0, диагоналийн дагуу үүссэн ирмэгийг 1 гэж тэмдэглэж зүүн дээд булангаас баруун доод булан хүртэлх тохиромжтой хамгийн урт дэд дарааллыг энэ граф дээр тодорхойлж болно. Энэ графыг

чиглэлтэй гэж үзвэл хамгийн урт замыг тооцоход графын хэмжээнээс хамаарч  $O(m \times n)$  хугацаанд олох боломжтой.

**2.2.5 Урьдчилан шахалт хийх:** Энэ арга нь бидний боловсруулсан гол аргачлал бөгөөд рекурсив хайлт хийхээс өмнө хоёр үгнээс бүтсэн L хоёр хэмжээт хүснэгт үүсгэж ижил үсэг харгалзсан нүдийг тэмдэглэнэ. Эндээс мөр баганад нэг ч тэмдэглэгээ ороогүй үсгийг хасаж L2 хүснэгт үүсгэж дараа нь рекурсив хайлт хийнэ[5].

Жишээ нь “сургууль”, “сүхбаатар”:

	с	ү	х	б	а	а	т	а	р
С	1	0	0	0	0	0	0	0	0
У	0	0	0	0	0	0	0	0	0
Р	0	0	0	0	0	0	0	0	1
Г	0	0	0	0	0	0	0	0	0
У	0	0	0	0	0	0	0	0	0
У	0	0	0	0	0	0	0	0	0
Л	0	0	0	0	0	0	0	0	0
Б	0	0	0	0	0	0	0	0	0

Хүснэгт 1. ХУДД-ын граф байгуулалт

Энэ хүснэгтийн мөр баганад нэг ч тэмдэглэгээ ороогүй хэсгийг хасахад хүснэгт 2 шинээр үүснэ.

	с	р
С	1	0
Р	0	1

Хүснэгт 2. Хүснэгт 1-д шахалт хийсний дараа

Хоёр үгийн уртыг  $m, n$  хэмжээтэй гэж үзвэл мөр бүрээс ядаж нэг тэмдэглэсэн нүд  $L2[i][j]=1$  олдох тул хүснэгтэд дараалсан жишилт хийхэд хамгийн ихдээ  $m \times n$  нөхцөл шалгах үйлдэл хийнэ.

**2.2.6 Дэд дарааллуудаас шүүлт хийх:** ХУДД олох алгоритм нь үндэс үг болон бичвэр үгнээс зөвхөн дэд мөрийг үүсгэдэг. Тиймээс дэд дарааллууд үүссэний дараагаар бичвэрт тохирох үгсийг ялгаж авах хэрэгтэй. Энд дараах 2 нөхцөлөөр шалгуур тавин шүүлт хийнэ.

1. Дэд мөр хамгийн урт байх
2. Үндэс үг болон дэд мөрийн зөрүү хамгийн бага байх.

Жишээ: Алдаатай бичвэр үг - “сутгууль”

Хүснэгт 3-т байгаа “саруул” гэсэн үг тохиромжтой мэт боловч дэд мөрийн урт нь “сургууль” гэсэн үгний дэд мөрөөс хоёроор бага байна. Тиймээс “сутгууль” гэсэн бичвэрийн үгийн үндсээр “сургууль” болно.

Үгийн сан	ХУДД	Дэд мөрийн урт	Зөрүү
Сар	с	1	2
Сам	с	1	2
Сал	сл	2	1
саруул	сруул	5	1
Сур	су	3	1
сурагч	суг	3	3
сургааль	сугль	5	3
сургууль	сугууль	7	1

Хүснэгт 3. Дэд дарааллуудаас шүүлт хийх

Хэрэв алдаатай бичвэр үг нь дагавар, нөхцөлөөр хувилсан үг байвал үгсийн сангаас олсон үндэс үгэнд тохирох нөхцөл дагавруудыг залгаж хувиргана. Шинээр үүссэн жагсаалтаас дахин ХУДД олох алгоритм хэрэглэж алдаагүй үгийг олно.

Жишээ: Алдаатай бичвэр үг - “сутгуулийн”

Үгийн сан	ХУДД	Дэд мөрийн урт	Зөрүү
сургууль	сугуул	6	2
сургуулиа	сугуули	7	2
сургуулийн	сугуулийн	9	1
сургуулийг	сугуулин	8	2
сургуульд	сугуул	6	3
сургуулиас	сугуули	7	3
сургуулиар	сугуули	7	3
сургуултай	сугуули	7	3

Хүснэгт 4. Хувилсан үгийн дэд дараалал

### III. ТЭМДЭГЛЭГЭЭ

Энэ ажлыг гүйцэтгэхэд мэргэжлийн үнэтэй зөвлөгөө өгч, чухал материалаар туслаж байсан, доктор (PhD) Д.Цэрэнпил агсаныг дурсахуй.

### IV. ДҮГНЭЛТ, ЦААШИД ХИЙХ АЖИЛ

Монгол хэлний алдаатай бичигдсэн үгийг зөв эсэхийг шалгах энэ аргачлал нь боловсруулалтанд орж байгаа үгийг бүх хувилбараар хувиргах, хувирсан үгнүүдээс шаардлага хангасан үгийг хурдан хайж олох шинэ хувилбарыг боловсруулсан.

Энд үгийг зөв хувилгах нийлэг алгоритмыг төгсгөлөг хоёртийн мод хэлбэрээр дүрсэлж, хэлзүйн дүрмийг алгоритмийн явцын шаардлагууд болгон тодорхойлсон. Хувираагүй үндэс болон язгууруудыг тодорхой атрибут бүхий үгийн сан болгон зохион байгуулсан.

Хувирсан үг зөв эсэхийг шалгахдаа NP-hard асуудалд шилждэг ХУДД олох аргыг сайжруулан нийлэг алгоритмтой хослуулан хэрэгжүүлсэн.

Цаашид энэ аргачлалаа улам боловсронгуй болгон төлгөлдөржүүлж бүтээгдэхүүн болгох, улмаар кирилжин монгол хэлний боловсруулалтанд ашиглах суурь ажил болсон.

#### НОМЗҮЙ

[1] D.Tserenpil, "A composing Algorithm for the To-Cyrillic Operations" – Acta Mongolica (The Center for Studies, NUM), т.7 (291), УБ., 2007, р.45-60.

- [2] Apostoli o, A. and Guerra, C., "The Longest Common Subsequence Problem Revisited" (1985). Computer Science Technical Reports. Purdue University Libraries.
- [3] Anna Gorbenko and Vladimir Popov, The Longest Common Subsequence Problem, Advanced Studies in Biology, Vol. 4, 2012, no. 8, 373 - 380
- [4] Batzolboo Bataa, Nyambaa Shimen, Finite-state automaton as a morphological model for the Mongolian language, The International Conference on Engineering (ICE2012), Mokwon University, Deajeon, Korea, May 2013, pp 87-90
- [5] Э.Авирмэд, "Кирилжин монгол бичгийн алдаа засах програм", Бакалаврын судалгааны ажил, Улаанбаатар хот, 2011 он