**UNIVERSITATEA ACADEMIEI DE ȘTIINȚE A MOLDOVEI**

**FACULTATEA ȘTIINȚE EXACTE**

**CATEDRA MATEMATICĂ ȘI INFORMATICĂ**

**UNGUR ȘTEFAN**

**ALGORITMIZAREA PROCESULUI DE TRANSLITERARE PENTRU ACTUALIZAREA TEXTELOR CHIRILICE ROMÂNEȘTI**

**TEZĂ DE LICENȚĂ**

**443.1 Informatică**

**Conducător științific: Gladei Anatol, lector superior**

**Autor: Ungur Ștefan,** **I3**

**Admis la susținere**

Șef catedră:

dr., conf. univ., Corlat Andrei

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

”\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_2016

**Chișinău – 2016**

**CUPRINS**

**ADNOTARE** 3

**ANNOTATION** 4

**LISTA ABREVIERILOR** 5

**INTRODUCERE** 6

1. PRINCIPII GENERALE, INTRODUCERE ÎN PROCESUL DE TRANSLITERARE8

1.1 Scurt istoric8

1.2 Perioadele de evoluție a limbii române10

1.3 Perioadele de evoluție al alfabetului român12

1.4 Principii generale ale transliterării14

1.5 Exemplu practic de transliterare lingvistică (din engleza în română)15

1.6 Aplicații practice ale transliterării19

1.7 Evaluarea metodei practice22

1.8 Digitizarea și recunoașterea textelor tipărite23

1.9 Recunoașterea textelor chirilice25

1.10 Procesarea textelor tipărite cu alfabet chirilic26

1.11 Procesarea textelor tipărite cu alfabet latin și litere adiționale 29

1.12 Procesarea textelor tipărite cu alfabete tranziționale sau cel slavon31

2. APLICAȚIA PRACTICĂ, ARHITECTURA ŞI DEZVOLTAREA PROIECTULUI34

2.1 Arhitectura aplicației34

2.2 Tehnologii utilizate39

2.3 Descrierea algoritmelor, regulilor și excepțiilor41

2.4 Exemplu practic de text transliterat 44

2.5 Împachetarea și distribuția 46

**CONCLUZII**47

**BIBLIOGRAFIE**48

**ADNOTARE**

*Numele și prenumele autorului*: Ungur Ștefan

*Denumirea tezei*: Algoritmizarea procesului de transliterare pentru actualizarea textelor chirilice românești

*Program de licență*: Informatică

*Structura lucrării*: adnotare, lista abrevierilor, introducere, 2 capitole, concluzii și bibliografie, 49 pagini de text formatul А4, 22 figuri și 11 referințe bibliografice.

*Cuvinte cheie*: transliterare, resurse lingvistice românești, alfabet slavon, recunoașterea textului, tehnologia limbajului, alfabet chirilic, transliterare, digitizare, alinierea textelor, Java, Swing.

*Scopul lucrării*: Realizarea și implementarea unui sistem de transliterare a textelor din grafie chirilică/slavonă în cea latină.

*Obiectivele lucrării*:

* elaborarea unui algoritm eficient de transliterare;
* algoritmizarea procesului dat, implementarea algoritmului într-un sistem software;
* studierea problemei dechirilizării limbii române;
* minimizarea erorii procesului de conversie de la grafia chirilică la cea latină;

*Materiale și metode:* Teza de licență abordează tematica implementării unui sistem unic de transliterare a mai multor tipuri de texte chirilice, ce sunt specifice mai multor perioade istorice. Așadar, realizarea unei aplicații software care va include mai mulți algoritmi de transliterare. De asemenea, realizarea unei interfețe grafice intuitive și ușor de utilizat. Baza teoretică și metodologică a tezei o constituie studierea lucrărilor fundamentale cu privire la specificul evoluției alfabetelor (slavon, chirilic și latin), analiza lucrărilor deja existente la tema dată. Implementarea algoritmilor realizați într-o unitate software pentru automatizarea procesului dat.

*Importanța lucrării*: Digitizarea și conservarea patrimoniului cultural istoric și lingvistic reprezintă o problemă prioritară din agenda digitală Europeană. UE a evidențiat necesitatea unui efort coordonat în acest domeniu și, respectiv, întreprinde vaste acțiuni în vederea impulsionării procesului dat, precum dezvoltarea bibliotecii virtuale „Europeana”, susținută prin rezoluția Parlamentului European și adoptarea Programului de lucru pentru activități culturale 2011-2014. Deci problema dechirilizării limbii române are o importanță primordială în păstrarea și dezvoltarea patrimoniului cultural și istoric românesc.

**ANNOTATION**

*Author Name*: Ungur Ștefan

*Thesis Title*: Algorithmization of the transliteration process in order to actualize romanian cyrillic texts.

*License Program*: Informatics

*Structure*: annotation, abbreviations, introduction, 2 chapters, conclusions and bibliography, 49 pages of A4 format, 22 figures and 11 bibliographic sources.

*Keywords*: transliteration, romanian linguistic resources, slavic alphabet, text recognition, language technologies, cyrillic alphabet, digitization, text alignment, Java, Swing.

*Thesis purpose:* Developing and implementation of a software that will automatically transliterate texts from slavic/cyrillic graphy to the latin one.

*Thesis objectives:*

* developing an efficient transliteration algorithm;
* implementing it into an software system;
* studying the decyrillisation of the romanian language;
* minimisation the error rate of the decyrillisation process.

*Materials and methods:* This thesis describes digitization of old Romanian texts, problems at their recognition, and motivates the necessity to create specific electronic resources mirroring the history of the standard Romanian language. So the main goal was developing a single software system that could transliterate more than one type of cyrillic romanian text (which are from different historical periods). Of course, first step was to develop the initial algorithms. The theoretical basis for this thesis was collected by studying another works with more or less appropriate goals.

*The importance of the thesis*: Evaluation and digitization of cultural and historic heritage is one of the main goals in the digital Europan agenda. Even if the process of heritage digitization needs many problems to be solved that refer to recognition, editing, translation, interpretation, it most certainly should be done. These problems became more complicated for Romanian as we need to consider the historic period when the source was printed, and we have more than one period. There is no doubt that decyrillisation of the romanian language is one of the most important steps in saving the romanian historic and cultural heritage.

**LISTA ABREVIERILOR**

*GUI* – graphical user interface, interfață grafică;

*DOM* – direct orthographic mapping, mapare ortografică directă;

*MT* – machine translation, traducere automată;

*RSSM* – Republica Sovietică Socialistă Moldovenească;

*CMUDict* – Carnegie Mellon University pronouncing dictionary;

*DEX* –dicționar explicativ al limbii române;

*OOV* – out of vocabulary, din dicționar;

*API* – application program interface;

*TTS* – text to speech;

*OCR* – optical character recognition, recuoașterea optică a caracterelor;

*CPU* – central processing unit, unitatea centrală de procesare;

**INTRODUCERE**

În această lucrare de licență voi prezenta detaliat demersul realizării unei aplicații de transliterare (convertare a textelor din grafia chirilică/slavonă în cea latină), și realizarea unui algoritm eficient pe baza căruia va fi realizată aplicația dată. De asemenea voi prezenta detaliat ceea ce trebuie cunoscut și ceea ce trebuie de luat în considerație atunci când se realizează o astfel de aplicație de convertare. Voi trece în revistă principiile de bază a alfabetelor și ortografiei textelor chirilice din 3 perioade istorice: chirilica sovietică (sec. XX), chirilica română slavonă (sec. XVIII) și chirilica tranzițională (sec. XIX). În lucrare voi evidenția principiile de bază a realizării unui algoritm pentru automatizarea transliterării dintr-o grafie în alta.

Pe parcursul lucrării voi pune accent atât pe importanța realizării unui astfel de algoritm, cât și pe crearea unei aplicații finale, cu interfață grafică accesibilă, un funcțional clar și performant. Voi defini caracteristicele importante la nivel vizual și tehnic pentru diferite tipuri de aplicații de convertare, clasificate după scop. Comportamentul și așteptările unui utilizator, care solicită astfel de aplicații, sunt corelate cu tipul și scopul acestor aplicații. Voi prezenta etapele de redactare a textelor inițiale chirilice și criteriile necesare pentru ca aplicația de convertare să se producă la nivelul maxim scontat. Deoarece aplicația doar convertează textele dintr-o grafie în alta, ea nu-și propune să corecteze greșelile de ortografie, de conținut sau de stil, astfel textul final, în grafie latină, va conține în totalitate astfel de greșeli, care au fost prezente în textul inițial chirilic. De asemenea dicționarul de cuvinte recunoaște doar cuvintele chirilice și latine scrise în forma lor exactă (corectă), așadar el nu va avea nici un efect asupra cuvintelor cu greșeli ortografice.

Aplicația de convertare nu se limitează doar la simpla transliterare a textelor propuse. O aplicație de tipul dat nu se rezumă doar la grafică sau design. Utilizarea pachetului de instrumente și funcționalități (dicționarul de cuvinte, lista de prefixe, alegerii standardului necesar etc.),face posibilă efectuarea aplicației de convertare la cel mai înalt nivel. În scopul unei mai bune funcționări a aplicației de transliterare, am elaborat o interfață grafică accesibilă și ușor de înțeles și un pachet de instrumente, care să fie ușor de aplicat. Orice utilizator apreciază, în primul rând, calitatea conținutului, iar calitatea unei aplicații e determinată atât de aspectele vizuale, cât și de cele tehnice. Am efectuat un studiu comparativ al mai multor aplicații de convertare, realizate atât peste hotare, cât și în țară. S-a analizat succint și s-a evaluat atât nivelul tehnic, cât și cel vizual, cu scopul de a face concluzii în vederea realizării proiectului de licență: aplicația de transliterare **ABConv**.

Pentru realizarea unei aplicații de acest gen și pentru realizarea unui algoritm de convertare automată dintr-o grafie în alta sunt necesare cunoștințe din diferite domenii: filologie, istorie, lingvistică etc. Realizarea unui algoritm corect și eficient este facilitată considerabil de aprofundarea cunoștințelor din domenii precum tipografia, istoria alfabetelor românești și a celor chirilice, de studiul ortografiei slavone, iar rezultatele devin net superioare.

Pentru a realiza exemplar și rapid transliterarea unui text din grafia chirilică/slavonă în cea latină, aplicația dată înglobează elemente tehnice și ortografice specifice acestora: regulile ortografice, corectitudinea textului, păstrarea spațiilor textului inițial, definirea unor cazuri speciale pentru caracterele ce au semnificație multiplă etc. Așadar, un text chirilic nu poate fi convertat mot-a-mot, acest proces trebuie realizat ținând cont de numeroase reguli lingvistice pentru a asigura păstrarea mesajului inițial al textului. De la programator se cere un grad înalt de cunoaștere în domeniul programării, dar și în domeniul filologiei, lingvisticii, istoriei și evoluției alfabetelor, ortografiei grafiei slavone etc. Toate aceste tehnologii și cunoștințe vor asigura corectitudinea transliterării.

Orice aplicație, atât desktop cât și web, la început nu este altceva decât un algoritm de procesare a datelor, de intrare și obținere a datelor, de ieșire necesare, iar fiecare algoritm la început se elaborează, testează, cizelează, pentru ca în final să fie implementat în aplicația dată. Interactivitatea aplicației se bazează pe aceste tehnologii, iar limitele impuse la nivel de implementare trebuie luate în considerare în conceperea unei aplicații. Mai mult decât atât, este important de luat în considerare și grupul țintă de utilizatori căruia se adresează aplicația. Chiar dacă aplicația nu este orientată spre un public larg de utilizatori, oricum trebuie de ținut cont de necesitățile și posibilitățile lor (vârsta, profesia etc.), cu scopul de a realiza interfața grafică și un pachet de instrumente cât mai eficient. În acest mod se poate determina mai ușor cum trebuie să arate designul, ce funcționalități trebuie să implementeze și ce grad de accesibilitate este necesar.

Totodată, prin efectuarea studiilor de marketing (studierea aplicațiilor asemănătoare, definirea grupului țintă de utilizatori etc.) GUI-ul și unele funcționalități ale aplicației pot fi considerabil schimbate, astfel încât să corespundă necesităților utilizatorilor.

**Capitolul 1** este format din 12 paragrafe și începe cu introducerea în istoria limbii române și perioadelor de evoluție al alfabetului român, apoi este formulată problema digitizării și recunoașterii textelor chirilice. În ultimele două paragrafe are loc formularea unor reguli de transliterare, mai întâi pentru chirilica sovietică (mijlocul sec. XX) iar apoi pentru grafia slavonă (sec XVIII-XIX).

**Capitolul 2** este format din 5 paragrafe și este total dedicat aplicației practice realizate, și anume arhitecturii, interfeței grafice, tehnologiilor utilizate, algoritmilor implementați și descrierii lor. În paragraful 2.4 este prezentat un exemplu de text vechi slavon (an. 1786) recunoscut cu ABBEYY FineReader și transliterat în grafia latină cu ajutorul aplicației practice date.

**1.** **PRINCIPII GENERALE, INTRODUCERE ÎN PROCESUL DE TRANSLITERARE**

**1.1. Scurt istoric**

De-a lungul timpului au fost propuse câteva tehnici de transliterare între două limbi, fiind orientate, îndeosebi, pe transliterarea ortografică a numelor proprii englezești în chineză, japoneză, coreeană sau arabă.

Knight şi Graehl, în 1997, au introdus o metodă de transliterare între japoneză și engleză, utilizând algoritmi de traducere bazați pe mașini cu stări finite, această metodă fiind adaptată de Stalls şi Knight în anul succesor pentru transliterare bidirecțională între engleză și arabă. Alte metode de transliterare au mai fost descrise de Jung et al. (2000), Meng et al. (2001), Virga și Khudanpur (2003). În lucrarea lor, Haizhou et al. (2004) clasifică metodele menționate mai sus ca fiind abordări ale transliterării bazate pe nivel fonetic. Ei propun o nouă tehnică, numită de către autori mapare ortografică directă (direct orthographic mapping sau DOM), cu alte cuvinte un model de transliterare pe bază de n-grame (secvențe de n litere consecutive care pot apărea în cuvintele unei limbi).

Experimentele realizate au fost focalizate asupra transliterării textelor românești tipărite cu grafie chirilică în cea română, fiind parte a studiului efectuat pentru modulul dedicat transliterării automatizate, integrat într-un sistem de sinteză a vorbirii pentru limba română.

Transliterarea nu este altceva decât reprezentarea caracterelor unei scrieri alfabetice sau silabice în caracterele alfabetului în care se face conversia. Transliterarea se face în principiu caracter cu caracter și redă grafia (nu pronunțarea) cuvintelor. Se utilizează în actele notariale, de stare civilă, în comunicațiile poștale, precum și în lucrări de istorie, geografie, cartografie, bibliografie, lingvistică etc.

Transliterarea se face după sistemele oficiale ale limbilor supuse transliterării sau după norme naționale sau internaționale de conversie. Ea nu depinde de normele ortografice ale limbii în care se transliterează. Standardele internaționale pentru conversia sistemelor de scriere urmăresc asigurarea unui sistem de convertire riguros, univoc și complet reversibil, fără ambiguități. Ele trebuie să permită schimbul internațional de informații prin comunicarea cu mijloace manuale, mecanice sau electronice a mesajelor scrise de oameni sau mașini, care să le poată transmite și reconstitui automat. Din acest motiv se face abstracție de considerente fonetice sau estetice și de uzanțele naționale. Reprezentările ce rezultată exclusiv în baza transliterării nu pot fi considerate mereu exacte conform uzanțelor fonetice ale limbii supuse conversiei. Totuși, cititorul care cunoaște limba convertită poate reconstitui în mod neechivoc grafia și, în anumite limite, pronunția originală.

Transliterarea respectă obiectul tradiției și convențiilor locale. De exemplu, în tradiția limbii române litera chirilică *х* se transliterează prin litera latină *h*, dar în tradiția țărilor anglofone aceeași literă se transliterează prin digrama *kh*. Astfel, în română toponimul Сахалин se transliterează Sahalin, iar în engleză ca Sakhalin. De asemenea, limba română redă ش arab, שׁ ebraic sau ш chirilic prin "ș", astfel că denumiri precum precum שבועות ,مراكش sau Шумeн, se transcriu respectiv în litere latine prin Șauia, Șavuot sau Șumen, pe când în țările anglofone sau francofone apar sub forme cu digrame: Shawia, Shavuot sau Shumen în engleză, respectiv Chaouia, Chavouot sau Choumen în franceză.

Desigur, în limba română, este preferabilă folosirea transliterării românești. Când este vorba, în schimb, de nume proprii străine deja scrise în litere latine în limbile țărilor respective, nume care așadar nu necesită să fie transliterate, tradiția este preluarea lor ca atare: Ouagadougou, Saskatchewan, Szczecin, cu indicarea pronunțării transcrisă între paranteze (Oagadugu, Sascatceoan, Șcețin). Aceste tradiții au fost în general respectate în cărțile și publicațiile apărute în România până la începutul deceniului 1980.

Cazuri particulare sunt:

* cele ale limbilor asiatice precum chineza, japoneza, coreeana sau limbile peninsulei indiene, pentru care se folosesc, atât în română, cât și în majoritatea limbilor folosind scrierea romanică, sisteme de transliterare specifice (de exemplu sistemul Pinyin pentru chineză);
* cel simplificat al limbii bulgare;
* cele două sisteme de transliterare folosite de-o parte și de alta a Prutului pentru denumirile, toponimele și patronimele din limbile rusă sau ucraineană: К, Ы, și Я se transliterează în România prin K, Â și IA (în dependență de conținut), iar în Republica Moldova prin C (CH înainte de E sau I), Î și EA sau IA (la fel în dependență de conținut).

Transliterarea și transcrierea se aseamănă, dar nu trebuie confundate. Prin transcriere se reprezintă sunetele vorbirii într-un anumit sistem de scriere, fără a ține seama de modul de notare a sunetelor în sistemul de scriere propriu al limbii respective, deci transcrierea este mai degrabă o metodă de notare fonetică (este metoda cea mai folosită în limbile folosind alfabetul chirilic). Transliterarea în schimb face conversia unui sistem de scriere în altul, nefiind condiționată de aspectul fonetic.

**1.2 Perioadele de evoluție a limbii române**

Istoria limbii române cunoaște două epoci în dezvoltarea sa. Prima se referă la formarea dialectului dacoromân, începând cu căderea Sarmisegetuzei (106 A.D.) până în secolul al XV-lea. Se utiliza alfabetul chirilic grație influenței masive a Bisericii Ortodoxe.

Epoca a doua de dezvoltare a limbii române literare (secolul XVI-XX) începe cu apariția primelor texte scrise în limba română și constituie rezultatul unei îndelungate și complexe evoluții. Procesul de unificare lingvistică este marcat de apariția Bibliei de la București (1688), care a condus ulterior la stabilirea a două mari etape în evoluția lingvistică.

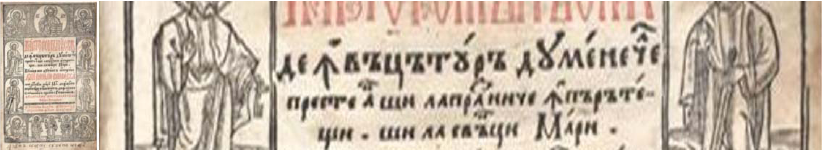
Etapa întâi începe cu apariția primelor texte literare românești și se încheie la începutul secolului al XVIII-lea. În cadrul acestei etape pot fi distinse 3 perioade:

* Anii 1532 și 1588, prima fază a limbii literare;
* Anii 1588-1656, faza consolidării principalelor variante ale limbii române literare (muntenească, moldovenească și sud-vest-ardeleană);
* Anii 1656-1715, faza influenței reciproce dintre variantele literare.

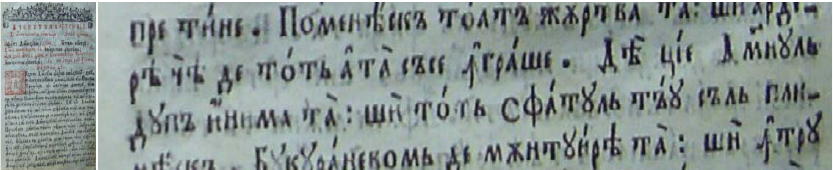
A doua etapă se întinde pe un interval între 1715 și 1960. Este epoca de consolidare a limbii unice supradialectale. Procesul de unificare a limbii române literare a cunoscut o evoluție lungă, în cursul a 4 perioade:

* Anii 1715-1780, momentul primei unificări, aproximativ în 1750;
* Anii 1780-1836, diversificarea lingvistică;
* Anii 1836-1881, constituirea principalelor norme ale limbii literare de astăzi;
* Ani 1881-1960, definitivarea formării normelor limbii române literare contemporane.

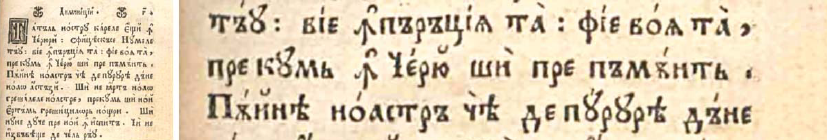
Ultima perioadă ne descrie consolidarea stilurilor limbii române literare. În 1904, prin modificările aduse ortografiei, se stabilesc definitiv bazele scrierii fonetice. păstrate, cu unele retușări ulterioare, până în prezent. Voi arăta în fig. 1-7 exemple de texte tipărite în diverse perioade din evoluție a limbii române.



**Figura 1**. Cazania lui Varlaam, Iaşi, 1643



**Figura 2**. Ceaslov, 1748

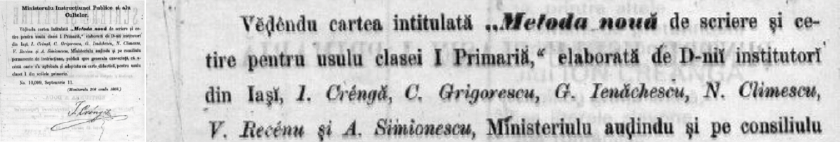


**Figura 3**. Татъл ностру. Acathist cu multe alease rugăciuni pentru evlaviia ﬁ eştecăruia creştin. Acum a treia oară tipărit. Blaj: Tipograﬁ ia Seminariului, 1786.



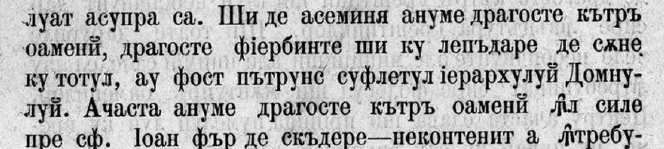
**Figura 4**. Letopisiţile Ţării Moldovii publicate pentru întăiaşi dată de Mihail Kogălniceanu. Tom I. Iaşii. La toate

libreriile. 1852

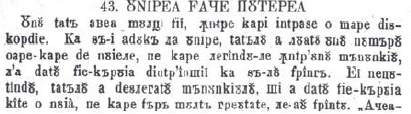
****

**Figura 5**. Metodă nouă de scriere şi cetire: Pentru usulu clasei I, Primariă / I. Créngă , C.Grigorescu, G. Ienăchescu,

Ed. a II-a. – Iassy: Tipograﬁ a H. Goldner, 1868. – 71 p

****

**Figura 6**. O pagină din revista „Луминъторюл”, 1908, Nr. 1

****

**Figura 7**. Una din variantele alfabetului tranziţional utilizat în alfabetul lui I. Creangă.

„Ш’ачел реже-ал поезией, вечник тынэр ши фериче,

Че дин фрунзе ыць дойнеште, че ку флуерул ыць зиче…”

**1.3 Perioadele de evoluție a alfabetului român**

În secolul al XVII-lea, tiparul românesc utiliza un alfabet chirilic cu 47 de litere, majoritatea dintre ele fiind împrumutate din alfabetul bisericii slavone. S-au adăugat câteva litere grecești, în mare parte pentru redarea adecvată a numelor proprii, dar și litere originale românești. De exemplu litera ꙟ utilizată pentru a reda prefixul (prepoziția) **în**, **îm**, sau litera **î** modernă la începutul cuvântului. Acest alfabet a fost utilizat la tipărirea *Cazaniei* lui Varlaam la Iași în anul 1643 (fig. 1). Primul abecedar românesc a fost tipărit în 1699 la Belgrad (Alba-Iulia), iar prima gramatică românească a fost tipărită în 1757 de Dimitrie Eustatievici.

Începând cu anul 1830 și până la adoptarea oficială a alfabetului român în 1862, nu exista un alfabet stabil, astfel în această perioadă au fost utilizate cel puțin șapte modificări ale așa-numitului „alfabet de tranziție”, chirilic-latin, care conținea atât litere latine, cât și litere chirilice (fig 4, 7). De exemplu, **e** - **e** (1830) - **з** (1846); **к** - k; **щ** - **шt**; s - **дз** - **dz** - **ɖ** (1846).

Utilizarea grafiei latine în România nu a influențat activitatea tipografică din Basarabia. După alipirea Basarabiei la Imperiul Rus în 1812, limba oficială la Chișinău devine rusă. În anul 1833 limba română a fost exclusă din circuitul oficial, dar a continuat să fie utilizată în activitățile eparhiale. Astfel, pe parcursul anilor 1867-1871 apărea versiunea română a monitorului eparhiei Chișinău tipărit cu caractere chirilice. Tipografia bisericească din Chișinău a fost sistată în perioada 1883-1890, procesul fiind reluat la începutul secolului XX.

Spre deosebire de alfabetul chirilic utilizat pentru scrierea limbii române din secolele XIV-XV până în anul 1862, alfabetul chirilic folosit în Republica Autonomă Sovietică Socialistă Moldovenească (RASSM) începând cu anii 1930 și, ulterior în Republica Sovietică Socialistă Moldovenească (RSSM) și Transnistria în prezent, este de fapt o adaptare a alfabetului chirilic rusesc. De menționat că în perioada 1932-1938 în RASSM a fost utilizat alfabetul latin. În Republica Moldova alfabetul chirilic a fost utilizat până în 1989.

Voi prezenta mai jos (tab. 1) perioadele de evoluție a alfabetului român începând cu *Cazania* de Varlaam. Pe lângă alfabet, există și alți factori care caracterizează evoluția limbii, precum ortografia și lexiconul.

|  |  |  |
| --- | --- | --- |
| **Tabelul 1**: Evoluția alfabetului român începând cu anul 1642 | |  |
| **România** | **Basarabia** | |
| 1642 – 1710 (alfabet chirilic) | | |
| 1710 – 1830 (alfabet chirilic modificat) | 1710 – 1814 (alfabet chirilic modificat) | |
| 1830 – 1862 alfabet tranzițional, mixt chirilic-latin) | 1814 – 1880 (alfabet chirilic bazat pe alfabetul rus și cel slavon bisericesc; ocazional alfabet tranzițional latin) | |
| 1862 – 1904 (alfabet latin) | 1880 – 1905 (n-a existat tipar românesc)  1905 – 1918 (alfabet chirilic bazat pe alfabetul civil rus) | |
| 1904 – 1960 (alfabet latin modificat) | 1919 – 1940, 1941 – 1944 (alfabet latin modificat)  1940 – 1941 (alfabet chirilic bazat pe alfabetul rus) | |
| 1960 – 1993 (alfabet latin modificat) | 1944 – 1989 (alfabet chirilic bazat pe alfabetul rus; apare litera **ӂ**) | |
| 1993 – prezent (alfabet modern român bazat pe alfabetul latin) | 1989 – prezent (alfabet modern român bazat pe alfabetul latin) | |

**1.4 Principii generale ale transliterării**

Transliterarea în prelucrarea limbajului natural a fost introdusă pentru a translata numele proprii dintr-o limbă în alta în situațiile în care cele două limbi folosesc un inventar fonetic incompatibil sau o ortografie total diferita. În această lucrare, propun o metodă pentru translatarea din grafia chirilică și slavonă în cea latină (dar care, ulterior, poate fi adaptată pentru orice altă pereche de limbi sau grafii), și prezint o aplicație care utilizează transliterarea în sinteza vorbirii, astfel automatizând procesul nemijlocit de conversie.

În traducerea automată (MT – machine translation) sunt multe situații în care întâlnim nume proprii ai căror echivalenți de traducere nu sunt cunoscuți. În cazul în care ortografia celor două limbi este asemănătoare, o practică frecventă este ca aceste cuvinte să rămână neschimbate. Acest lucru nu este posibil însă, dacă una dintre limbi folosește o ortografie total diferită de cealaltă (de exemplu, traducerea din engleză într-una din limbile arabă, japoneză, rusă, chineză sau bulgară).

Sistemele de sinteză a vorbirii pornind de la text au sarcina de a sintetiza vocea pornind de la un text arbitrar, se confruntă cu următoarea problemă: pentru textele care conțin cuvinte sau nume proprii provenind din alte limbi nu se poate aplica direct transcrierea fonetică folosind aceleași reguli de transcriere specificate manual sau învățate automat pentru limba pe care a fost proiectat sistemul. O soluție pentru rezolvarea acestei probleme este introducerea unor pachete suplimentare de reguli în vederea obținerii transcrierii fonetice din diferite limbi sursă. Însă nu toate limbile au același pachet fonetic, iar lexicoanele străine necesită adaptări pentru a fi corelate cu limba sau grafia țintă.

În cadrul acestei lucrări, propun o abordare diferită a transliterării, în care se utilizează un transliterator pentru adaptarea de la o limbă la alta, înainte de a aplica transcrierea fonetică. Aceasta din urmă se obține folosind același pachet de reguli utilizat pentru cuvintele native (ale grafiei țintă). În cazul limbii române, datorită ortografiei preponderent fonetice, este necesar ca acuratețea obținută prin transliterare, urmată de transcrierea fonetică aplicată pentru grafia latină să fie comparabilă cu acuratețea obținută în situația în care s-ar aplica reguli de transcriere fonetică direct pentru chirilică, urmând să se facă ulterior o adaptare la nivel fonetic între cele două grafii. Practic, acuratețea globală este limitată în ambele cazuri de performanțele sistemelor de transcriere fonetică.

**1.5 Exemplu practic de transliterare lingvistică (din engleză în română)**

În cadrul lucrării mi-am propus că realizez, în mod practic, transliterarea atât dintr-o grafie în alta, cât și dintr-o limbă în alta. Spre a ușura condiția experimentului am considerat limba-sursă ca fiind cea engleză, deoarece face parte din aceiași familie lingvistică ca și limba română, și deoarece materiale legate de conversia din limba engleză în cea română sunt foarte multe și ușor de accesat. Așadar, algoritmul de transliterare din engleză în română va necesita un corpus de antrenare format din cuvinte scrise în limba sursă (în cazul dat engleză) asociate cu transliterările lor corespunzătoare în limba țintă (în cazul dat limba română).

Faptul că limba română are o ortografie preponderent fonetică a permis să folosesc transcrierea fonetică a cuvintelor englezești ca pivot pentru tehnica semi-automată pe care am folosit-o în crearea corpusului de antrenare. Am ales ca lexicon de plecare pentru transcriere fonetică în limba engleză CMUDict. Pe lângă cuvintele uzuale, acest lexicon conține un număr mare de nume proprii, abrevieri și cuvinte adaptate la limba engleză, care sunt provenite din arabă, germană, franceză, poloneză, precum:

* Italiană: braggiotti, castelli, castelluccio
* Germană: aachen, abbenhaus, schlender, schlenker
* Poloneză: zawistowski.

Aceste cuvinte ar încurca procesul de transliterare, deoarece conversia lor în foneme nu se poate obține cu ajutorul regulilor standard. Soluția pentru a învăța doar reguli omogene a constat în filtrarea CMUDict prin alegerea unui set de cuvinte uzuale din limba engleză pe baza cărora a fost generat lexiconul de transliterare. Folosind apoi datele din CMUDict, a avut loc generarea transcrierii fonetice pentru aceste cuvinte și am folosit un set de reguli pentru a trece de la fonemele limbii engleze la litere și/sau grupurile de litere din alfabetul românesc (mai detaliat descris în Tabelul 2).

Pentru limbile fără ortografie fonetică sunt necesari doi pași suplimentari fată de cei prezentați anterior. Primul pas constă în maparea dintre fonemele specifice pentru limba sursă și fonemele care apar în limba destinație. Al doilea pas implică trecerea din forma fonetică a cuvintelor înapoi la forma ortografică a acestora, de data aceasta folosind un pachet de reguli specific pentru limba țintă. Ultimul pas se poate realiza folosind metode automate, dar, pentru rezultate bune, este necesară o iterație suplimentară care constă în validarea manuală a rezultatelor obținute automat.

**Tabelul 2**. Reguli de conversie

|  |  |  |  |
| --- | --- | --- | --- |
| ***Engleză*** | ***Cuvînt*** | ***Transcriere fonetică*** | ***Transliterație*** |
| AA | odd | **AA** D | **a**d |
| AE | at | **AE** T | **e**t |
| AH | hut | HH **AH** T | h**a**t |
| AO | ought | **AO** T | **o**t |
| AW | cow | K **AW** | c**au** |
| AY | hide | HH **AY** D | h**ai**d |
| B | be | **B** IY | **b**i |
| CH | cheese | **CH** IY Z | **ci**z |
| D | dee | **D** IY | **d**i |
| DH | thee | **DH** IY | **z**i |
| EH | Ed | **EH** D | **e**d |
| ER | hurt | HH **ER** T | h**ăr**t |
| EY | ate | **EY** T | **ei**t |
| F | fee | **F** IY | **f**i |
| G | green | **G** R IY N | **g**rin |
| HH | he | **HH** IY | **h**i |
| IH | it | **IH** T | **i**t |
| IY | eat | **IY** T | **i**t |
| JH | gee | **JH** IY | **g**i |
| K+(E/I) | key | **K** IY | **ch**i |
| K | call | **K** AO L | **c**ol |
| L | lee | **L** IY | **l**i |
| M | me | **M** IY | **m**i |
| N | knee | **N** IY | **n**i |
| NG | ping | P IH **NG** | pi**ng** |
| OW | oat | **OW** T | **ă**ut |
| OY | toy | T **OY** | t**o**i |
| P | pee | **P** IY | **p**i |
| R | read | **R** IY D | **r**id |
| S | sea | **S** IY | **s**i |
| SH | she | **SH** IY | **ș**i |
| T | tea | **T** IY | **t**i |
| TH | theta | **TH** EY T AH | **t**eta |
| UH | hood | HH **UH** D | h**u**d |
| UW | two | T **UW** | t**u** |
| V | vee | **V** IY | **v**i |
| W | we | **W** IY | **u**i |
| Y | yield | **Y** IY L D | **i**ild |
| Z | zee | **Z** IY | **z**i |
| ZH | seizure | S IY **ZH** ER | si**j**ăr |

Trebuie menționat că întregul proces de transliterare se face cu pierdere de informație, atât din cauza metodelor statistice folosite, cât și pentru că nu toate fonemele au echivalent direct în limba română. Pe baza contextului, unele litere din alfabetul român pot avea o pronunție diferită fată de cea dorită. Pentru cazul dat acest efect nu constituie o problemă, deoarece cuvintele sună „natural” pentru un vorbitor nativ de limba română în momentul în care sunt sintetizate. În cazul căutării pe bază de percepție, pentru a diminua efectele nedorite generate de această pierdere de informație, a fost folosită transliterarea „forward-transliteration” în locul „backward-transliteration” Diferența dintre aceste două metode se rezumă la faptul că prima implică inițial convertirea din engleză în română, iar a doua invers, din română în engleză. Deși căutarea se face folosind cuvinte scrise în română (ceea ce ar implica „backward-transliteration”), probleme la pasul dat nu au fost întâlnite.

Problema transliterării poate fi reformulată în modul următor: realizarea unui set de reguli, care pornind de la un șir de simboluri sau caractere ce aparțin alfabetului sursă (cuvintele ce trebuie transliterate nemijlocit), rezultă în obținerea unui șir de simboluri (caractere) ce aparțin alfabetului destinație, astfel încât aplicând setul de reguli pentru transcrierea fonetică specifică fiecărei limbi pe ambele șiruri, atât cel de intrare, cât și cel de ieșire, asemănarea între sunetele obținute să fie maximă. Metodele bazate pe manipulare ortografică directă au o acuratețe mai bună decât cele din prima categorie, ceea ce m-a determinat să aleg o abordare de tip DOM.

Transcrierea fonetică este o problemă asemănătoare cu cea a transliterării propriu-zise. Diferența dintre aceste două se formulează în modul următor: în cazul transcrierii fonetice se caută și se formulează un pachet de reguli pentru a trece din literele unui cuvânt anumit în simbolurile folosite pentru reprezentarea fonetică a acestuia, pe când transliterarea implică căutarea unei mapări către ortografia limbii-țintă. Așadar, metoda dată folosește un clasificator de tipul *Maximum Entropy*, care parcurge și asociază etichete fiecărui caracter dintr-un cuvânt pe baza unei serii de trăsături extrase din contextul lexical al literei respective.

Secvența de etichete sau, cu alte cuvinte, pachetul de reguli obținut pentru o secvență de caractere a unui cuvânt constituie transcrierea fonetică a acestui cuvânt. Dacă notăm cu **l** litera curentă, cu **li** litera aflată la distanta respectivă de lungime **i** fată de litera curentă și notăm cu **p-1** eticheta anterioară, atunci obținem o legitate a trăsăturilor, formulate în modul următor:

* l-1, l – litera curentă și litera anterioară;
* l-2, l-1, l – litera curentă și două litere anterioare;
* l, l+1 – litera curentă și litera succesoare;
* l, l+1, l+2 – litera curentă și două litere succesoare;
* l-1, l, l+1 – litera curentă împreună cu literele imediat învecinate;
* p-1 – eticheta emisă pentru litera predecesoare.

Acuratețea medie obținută cu ajutorul unei astfel de combinații de trăsături este în jur de 90% pentru cuvintele din afara vocabularului ce sunt transcrise corect fonetic în limba română și în jurul a 60% pentru cuvintele luate din CMUDict (deci din afara vocabularului), ce sunt transcrise corect fonetic în limba sursă (în cazul dat engleza). Trebuie neapărat menționat faptul că acuratețea de 60% pentru CMUDict este destul de bună în comparație cu alte metode propuse de alți autori pentru transcrierea fonetică. Acele metode au o acuratețe ce poate varia între valorile 57% şi 62%. Performanta maximală obținută pentru acest lexicon este de 71% și a fost obținută folosind algoritmul *Marginal Infused Relaxed Algorithm*.

Reieșind din cele enumerate mai sus, am decis să utilizez aceeași metodă de transliterare şi pentru convertarea textelor din limba engleză în română, antrenând astfel același clasificator, dar de data aceasta pe baza unui lexicon de transliterare. Ținând cont de trăsăturile menționate, clasificatorul atribuie fiecărei litere din context un eticheta ce reprezintă un simbol nemijlocit, o mulțime de simboluri din alfabetul destinație sau mulțimea vidă.

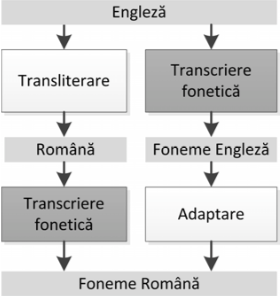
Etichetele obținute în acest mod, utilizând lexiconul de antrenare și folosind alinieri automate ale unor litere din cuvintele alfabetului sursă cu literele corespunzătoare din alfabetul destinație, au determinat un nivel înalt de precizie al algoritmului obținut.

Am cercetat și evaluat transliterarea din limba engleză în română, folosind o metodă numită ten-fold. Setul de date utilizat la antrenare a fost împărțit în 10 submulțimi de mărime egală și a fost testată acuratețea sistemului obținut pe toate submulțimile în parte, astfel antrenând sistemul pe celelalte 9 și calculând procentul de acuratețe la nivel de cuvânt. Rata acurateței a fost calculată în modul următor: numărul de cuvinte transliterate în mod corect fiind raportat la numărul total de cuvinte transliterate. Chiar dacă un cuvânt transliterat are o literă greșită, atunci acesta se atribuie la numărul de erori. Acuratețea sistemului dat, ca o medie a acelor 10 validări, a fost de 78%. Trebuie neapărat de menționat că rezultatul a fost raportat la cuvintele din afara vocabularului utilizat la antrenare. Din motiv că limba română are o ortografie preponderent fonetică, s-ar fi putut aștepta ca rata acurateței în cazul dat să fie nemijlocit asemănătoare cu rezultatul obținut la transcrierea fonetică. Creșterea cu 9% al acurateței în cazul transliterării se datorează faptului că antrenarea a fost făcută pe cuvintele doar din limba engleză (fără oricare cuvinte străine sau alte abrevieri ce se conțin în CMUDict). Mai mult ca atât, fiind vorba de o oarecare mapare ortografică dintre cele două limbi, a fost redus numărul etichetelor care se atribuiau unei litere sau unui grup de litere direct în momentul clasificării.

**1.6 Aplicații practice ale transliterării**

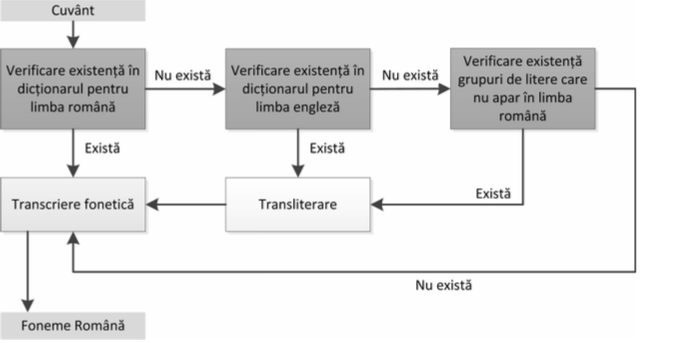
După cum a fost menționat mai sus, se întâlnesc mai multe situații în care textul, ce trebuie sintetizat, conține mai multe cuvinte provenite din alte limbi, care, evident, nu pot fi procesate corect, dacă vom utiliza același pachet de reguli, ca și la cuvintele specifice limbii-țintă pentru care este destinat sistemul (în cazul dat limba română). În situația dată, o abordare evidentă va fi folosirea unor seturi diferite de reguli de transcriere fonetică pentru aceste cuvinte. Aceste reguli vor fi specifice limbii din care ele fac parte, totodată fiind necesară și o adaptare ulterioară la nivelul fonetic. În cadrul acestei lucrări, propun o abordare puțin diferită, ce constă în folosirea unui transliterator pentru generarea “pseudo-cuvintelor” native. Așadar, transcrierea fonetică se va face folosind reguli specifice limbii române și nu va mai fi necesară adaptarea rezultatelor la nivel fonetic. Totuși există unele situații în care, fiind dat un context lexical, unele litere din limba română vor genera sunete diferite de cele care au fost intenționate. Cu toate acestea, cuvintele vor suna cât se poate de clar pentru orice vorbitor nativ de limba română și nu vor prezenta dificultăți în ceea ce privește inteligibilitatea. Diferența între aceste două abordări majore este subliniată în Figura 8 de mai jos:

* metoda bazată pe manipularea fonetică se aplică atunci când sunt cunoscute seturile de reguli pentru transcrierea fonetică (conversia din litere in sunete) pentru ambele limbi implicate in proces;
* metoda bazată pe manipularea ortografică nu necesită cunoașterea setului de reguli folosite pentru transcrierea fonetică a limbii- sursă, ci doar pentru limba-destinație.

**Figura 8**. Conversia cuvintelor scrise în engleză pentru sinteza vorbirii

Dificultatea cu care se confruntă ambele metode pentru procesarea cuvintelor străine este alegerea corectă a cazurilor în care sistemul trebuie să aplice transliterarea cuvintelor din limba-sursă în limba-țintă. Principala modalitate de a verifica, dacă un cuvânt trebuie sau nu transliterat, este folosirea lexiconului care a fost generat anterior. Este evident că, dacă un cuvânt se regăsește în tabela de transliterare, atunci el este și un candidat bun pentru procesul dat. Totuși este necesar de verificat, dacă acesta există și în inventarul de cuvinte al limbii- țintă. Spre exemplu, nu putem spune cu exactitate dacă transliterarea este necesară pentru un cuvânt de tipul “minus” sau “bat”, deoarece ele au aceeași semantică și ortografie atât în limba română cât și în cea engleză. Acest tip specific de cuvinte vor face parte din lista excepțiilor și, preferabil, stocate într-un oarecare fișier separat și, ca rezultat, vor rămâne neschimbate. În cazurile în care un cuvânt nu se va regăsi nici în inventarul de cuvinte cunoscute pentru limba- sursă și nici în cel pentru limba-țintă, va fi greu de precizat dacă acesta trebuie neapărat transliterat sau nu.

O metodă clasică de a rezolva aceste situații ar fi folosirea unor indicii lexicali pentru a decide care este limba din care provine cuvântul dat. Spre exemplu, dacă luăm limba sursă ca cea română atunci există anumite grupuri de caractere ce sunt foarte rare sau chiar nu pot exista în limba română. Deci toate cuvintele care conțin litera ‘y’ sau grupul de litere “ck” vor fi candidați buni pentru transliterare. Astfel, putem genera o listă de perechi de câte trei sau mai multe litere consecutive posibile în limba română, folosind DEX-ul. Atunci când sistemul va întâlni un cuvânt necunoscut (pentru ambele limbi procesate), va avea loc testarea dacă fiecare grup de litere consecutive din cuvânt se regăsește în lista generată anterior. Orice cuvânt care va conține o combinație de litere care nu este specifică pentru limba română va fi transliterat automat. Celelalte cuvinte vor fi lăsate neschimbate. Schema este descrisă în Figura 9.



**Figura 9**. Detectarea cuvintelor trebuie transliterate

După cum a fost menționat mai sus, căutarea pe bază de percepție reprezintă o modalitate de a găsi scrierea corectă a unui cuvânt dintr-o limbă-sursă (de regulă limbă străină, precum cea rusă engleză, germană, franceză etc.) în funcție de modul în care acest cuvânt este perceput (modul în care “sună” cuvântul) pentru un vorbitor nativ, în cazul dat de limba română. Spre exemplu, presupunem că știm nimic altceva despre un oarecare oraș cu excepția faptului că denumirea lui sună în modul următor: „ianţiau”. Așadar, nu avem nici o informație cu privire la tara în care se află orașul dat sau vreo altă informație despre ortografia pe care ar trebui să o folosim pentru a găsi mai multe date despre locație. În acest caz putem folosi căutarea pe bază de percepție pentru a obține scrierea exactă a denumirii locației pur și simplu prin tastarea cuvântului așa cum este el perceput în limba nativă. Deci un vorbitor nativ de limba română ar introduce cuvântul respectiv „ianţiau”, care reprezintă cea mai apropiată formă ortografică din limba sa, iar rezultatul ar fi “燕郊” – o localitate ce se află la nordul Chinei.

Această metodologia are și câteva neajunsuri:

* există mai multe cazuri în care unele scrieri diferite se pronunță la fel (omofone);
* părerea unei anumite persoane care nu este vorbitoare nativă de limba sursă despre cum ar trebui să fie scris cuvântul în limba sa nativă nu poate fi absolut exactă, deoarece nu toate limbile au același inventar fonetic iar regulile de conversie de la forma ortografică la cea fonetică sunt destul de complexe în unele situații.
* Așa cum a fost menționat de Knight şi Graehl (1997) transliterarea inversă (*backward-transliteration*) nu are aceeași flexibilitate ca și transliterarea directă (*forward-transliteration*), deoarece pierderea de informație este de două ori mai mare atunci când se transliterează̆ înainte și înapoi intre acele două limbi.

Când folosim căutarea bazată pe percepție ar fi bine de a antrena sistemul să translitereze între limba nativă (limba sursă în care nemijlocit s-a efectuat căutarea) și toate limbile țintă, alegând astfel cea mai bună variantă pe baza unei funcții de similaritate între șiruri de caractere. Totodată, această metodă este predispusă și unei serii de erori, precum ar fi faptul că o reprezentare fonetică "percepută" a unui cuvânt poate corespunde mai multor forme ortografice și, desigur, pierderea informației din cauza incompatibilității pachetelor fonetice ale celor două limbi.

Pentru a compensa astfel de erori am propus o abordare puțin diferită, și anume: toate cuvintele din limba țintă să fie transliterate în limba nativă de căutare. Când se efectuează căutarea, comparăm transliterația curentă (dată de utilizator) cu toate transliterațiile din baza de date folosind distanta Levenshtein.

**1.7 Evaluarea metodei practice de transliterare**

Pentru a valida această metodă căutare am realizat un alt corpus de test, compus doar din nume de orașe din SUA. Corpusul dat conține 420 de intrări selectate la întâmplare (deci nu are nimic în comun cu corpusul de transliterare, astfel niciun nume propriu nu a fost păstrat în CMUDict). Alegerea dată s-a bazat pe faptul că o astfel de metodă și-ar găsi foarte ușor locul într-un sistem de navigație, sau asistent de călătorie.

După ce am selectat aceste nume, am folosit Google Speech API pentru a sintetiza fiecare cuvânt şi am rugat un număr de 5 persoane să asculte înregistrările și să scrie cuvintele în română așa cum le aud. Fiecare persoană a avut posibilitatea să asculte același cuvânt nu mai mult de 3 ori.

Cuvintele ce aparțin corpusului de test nou creat au fost procesate conform metodologiei de căutare pe bază de percepție ce a fost prezentată anterior. La calcularea acurateței sistemului s-a obținut 98.38% (ceea ce implică că marea majoritate a cuvintelor au fost identificate corect).

În lucrarea de față am prezentat un exemplu de transliterare din engleză şi română, care, cu câteva adaptări specifice, ar fi posibil de aplicat și pe orice alte perechi de limbi. A fost creat un corpus de antrenare pentru transliterare ce poate fi obținut într-un mod semiautomat, deci fără efort pentru limbile cu ortografie fonetică Acuratețea de 78% a transliterației este raportată la cuvinte din afara vocabularului, deci OOV (sau out of vocabulary). Totodată nu toate cuvintele străine sunt total necunoscute și, chiar dacă uneori apar erori la transliterare pentru unele cuvinte din afara vocabularului, ele sunt evaluate în sinteza vorbirii în defavoarea formei lor directe. Ca parte din dezvoltarea sistemului *text-to-speech* românesc dat, ar fi de dorit să fie extins lexiconul de transliterare la franceză și, ulterior, germană. Rezultatul obținut utilizând căutarea pe bază de percepție arată că majoritatea motoarelor de căutare şi asistenții de călătorie vor beneficia de pe urma unui asemenea instrument. Căutarea după percepție ar putea îmbunătăți experiența utilizatorului de internet, și în același timp, atragerea atenției la corectarea greșelilor de ortografie bazată pe similitudini fonetice între cuvinte ar putea considerabil îmbunătăți procesul de corectare ortografică.

**1.8 Digitizarea și recunoașterea textelor tipărite**

Digitizarea și conservarea patrimoniului cultural istoric și lingvistic reprezintă o problemă prioritară din agenda digitală Europeană. Menționez recomandările Comisiei Europene „Privind digitizarea și accesibilitatea online a materialului cultural și conservarea digitală” din 27 octombrie 2011.

Dezideratele principale ale politicii culturale pentru zonele unde se vorbește limba română țin de studierea, valorificarea și digitizarea patrimoniului cultural-istoric. Procesul de digitizare a patrimoniului necesită soluționarea unui șir de probleme legate de recunoașterea, editarea, traducerea, interpretarea, circularea și recepționarea textelor tipărite atât în limba română cât și în alte limbi moderne. Soluționarea acestor probleme pentru patrimoniul istoric și lingvistic românesc se confruntă cu dificultăți și aspecte specifice: un număr mare de perioade în evoluția limbii, un număr relativ mic și foarte dispersat de resurse depozitate, o mare diversitate de alfabete folosite la tipărirea lor, în particular câteva „alfabete de tranziție” chirilic-latine. Dificultățile în digitizarea și conservarea acestui tezaur țin de recunoașterea corectă a literelor chirilic-latine, dar și de inexistența unui lexicon adecvat perioadei de tipărire a resursei. O soluție pentru problema lexiconului ar fi alinierea la normele lingvistice contemporane ale textelor vechi.

Istoric, limba română a parcurs o cale lungă și bogată de dezvoltare. Există studii care explică apariția foneticii și ortografierii caracteristice etapelor concrete de evoluție a limbii, care sunt necesare atât pentru determinarea alfabetului, cât și a literelor specifice. Cunoașterea acestor legități ne permite să construim resurse lingvistice utilizând un instrument special elaborat pentru o perioadă istorică concretă.

Prima carte tipărită pe teritoriul românesc a fost *Liturghierul slavon*, îngrijit de către ieromonahul Macarie în anul 1508, iar prima carte tipărită în limba română a fost *Catehismul Românesc* al diaconului Coresi, apărut la Brașov în anul 1535.

Biblioteca Națională a Republicii Moldova deține o colecție de aproximativ 21.000 cărți vechi și rare. Circa 20 de cărți din această colecție sunt tipărite în limba română, în Basarabia (Chișinău și Dubăsari), utilizând alfabetele chirilic și tranzițional. Bibliotecile publice din Sankt Petersburg dețin importante mostre de carte românească veche (secolele XVI-XIX). Dintre cele 66 de titluri incluse, spre exemplu, în *Catalogul edițiilor chirilice ale slavilor de sud și ale românilor*, 45 de volume revin slavilor de sud, iar 21 de volume - țărilor românești.

Studiile existente explică aspectele legate de dezvoltarea componentelor principale ale limbii: alfabet, lexicon, ortografie cu referire la etapele specifice din evoluția limbii. Această informație este utilă pentru a crea resurse și instrumente lingvistice racordate la anumite perioade din istoria limbii. Ținând cont de particularitățile fiecărei perioade, vom propune o tehnologie pentru crearea acestor componente. În particular, vom studia problema de digitizare a textelor tipărite cu caractere chirilice în Republica Sovietică Socialistă Moldovenească (RSSM) în perioada 1967-1989.

Lucrarea prezintă un proiect pe termen lung, care abia începe. Pe parcurs ne vom conduce de principiul „din prezent în adâncul secolelor”.

**1.9 Recunoașterea textelor chirilice**

Procesul de digitizare și de recunoaștere pentru manuscrise este destul de complicat, deoarece necesită efectuarea unor operații suplimentare, de exemplu ajustarea contrastului, „curățirea imaginii”, segmentarea textului. De asemenea, trebuie elaborați algoritmi speciali de recunoaștere și lexicoane specializate. Procesul de digitizare și recunoaștere e constituit din următoarele etape (fig. 10):

* Digitizarea (scanarea) textului pentru obținerea copiei electronice grafice;
* Recunoașterea cu metode standardizate, adică utilizarea nemijlocită a OCR, sau prin instruirea lui. În caz contrar, se vor folosi proceduri ale Inteligenței Artificiale, așa-numitul proces de conversie. Transliterarea textului se va efectua ținând cont de literele specifice utilizate în textul inițial.
* Verificarea textului recunoscut se produce utilizând resursele lingvistice reutilizabile specializate pentru perioada respectivă.

|  |
| --- |
| C:\License\LICENSE_IMG\Figura_08.png Figura 10. Etapele tehnologice de recunoaștere a textelor tipărite |

Digitizarea textelor constă în scanarea lor și obținerea variantei electronice în formă de

imagine. Pentru recunoașterea textelor din imagine se aplică OCR. Sistemele standard OCR utilizează diferite metode de recunoaștere a textelor. Am cercetat posibilitățile a două sisteme: IRIS și ABBYY FineReader. Rezultatele experiențelor de recunoaștere a textului tipărit în secolul al XIX-lea sunt expuse în continuare. Am determinat că sistemul IRIS, în procesul de instruire, nu poate selecta orice fragment din imagine textului și de aceasta acest sistem nu satisface scopul de recunoaștere a textului tipărit cu scrisul vechi român. Așadar, sistemul IRIS nu a mai fost utilizat în scop practic, ci doar ca un model comparativ.

În continuare se vor folosi tehnici de recunoaștere a formelor pentru identificarea individuală a caracterelor unei pagini de text, inclusiv și semnele de punctuație, spațiile și sfârșitul de linie. Textul recunoscut se va prezenta în final ca un fișier editabil.

Transliterarea este un proces strict individual ce depinde de perioada examinată. În funcție de textul inițial, se vor utiliza programe care conțin informație despre caracterele specifice întâlnite în text. Transliterarea presupune stabilirea unei relații bidirecționale univoce între două sisteme de scriere astfel, încât un cunoscător să poată reconstitui textul original din varianta transliterată.

Verificarea textului se efectuează cu aplicații special elaborate, care utilizează resursele reutilizabile specifice pentru perioada istorică a textului tipărit. Totodată, cuvintele noi obținute se vor introduce în lexiconul corespunzător.

**1.10 Procesarea textelor tipărite cu alfabet chirilic**

Perioada inițială de utilizare a alfabetului chirilic în Republica Autonomă Sovietică Socialistă Moldovenească (RASSM) se referă la anii 1924-1940 și se asociază cu utilizarea unui lexicon foarte specific, caracterizat de:

* utilizarea cuvintelor rusești (de exemplu **совет**, **указ**, **словарь**) în locul echivalentelor românești (**consiliu, decret, dicționar**);
* excluderea neologismelor românești, ele fiind considerate „burghezisme”;
* utilizarea lexiconului local (Transnistrean);
* introducerea unor neologisme auto-inventate pentru unele noțiuni abstracte neatestate în limbajul basarabean. De exemplu, **амувремник** (amuvremnic) în loc de **contemporan**;
* utilizarea particularităților accentului locar (Transnistrean), de exemplu **ди** (**di**) în loc de **de**, **мержи** (**merji**) în loc de **merge**, **сунити** (**suniti**) în loc de **sunete** etc.

Mă voi referi în continuare le perioada 1967-1989 de utilizare a alfabetului chirilic în Republica Sovietică Socialistă Moldovenească (RSSM). Pentru procesarea OCR a textelor apărute în această perioadă, este necesar să fie instruit sistemul OCR pentru a recunoaște litera adițională **ӂ** și pentru a crea lexiconul respectiv. Existența unui lexicon caracteristic acestei perioade ar permite automatizarea procesului de verificare și validare a cuvintelor recunoscute prin procedeele expuse mai sus. Acest lexicon poate fi creat:

* manual,
* prin transliterarea cuvintelor românești scrise cu caractere latine în varianta corectă scrisă cu caractere chirilice;
* prin alinierea variantelor de text tipărite în paralel cu caractere latine și caractere chirilice.

Prin transliterare vom înțelege transcrierea unui cuvânt din limba română în forma echivalentă scrisă cu caractere chirilice și conform normelor lingvistice acceptate în perioada 1967-1989 în RSSM.Metoda transliterării s-ar potrivi ideal în cazul dacă se reușește formalizarea tuturor regulilor de transcriere. Un studiu prealabil arată că acest proces este anevoios și nu poate fi automatizat în totalitate din cauza iregularităților legate de discordanța dintre fonologia, morfologia și sintaxa limbii române și normele lingvistice acceptate în RSSM. Parțial acest proces poate fi automatizat implicând elemente de formalizare a regulilor de transcriere, de intervenție manuală și de aliniere.

Dificultăți evidente apar la transliterarea cuvintelor de proveniență străină. Dacă în limba română aceste cuvinte se scriu, de regulă, ca și în limba originală, atunci transcrierea lor cu caractere chirilice se face conform pronunțării. De exemplu, **design** - **дизайн**, **cowboy** - **ковбой**, **watt** - **ват**, **charleston** - **чарлстон**. Aceste cuvinte pot fi transliterate doar în regim manual.

Pentru lexiconul original românesc procesul respectiv poate fi parțial automatizat. În acest scop s-au stabilit reguli de transcriere a literelor și îmbinărilor de litere. Exemplu de aceste reguli sunt prezentate mai jos:

* Reguli de transcriere „literă → literă”. De exemplu, ***a****→****а****,* ***ă****→****э****,* ***b****→****б****,* ***d****→****д****,* ***f****→****ф****,* ***l****→****л****,* ***m****→****м****,* ***n****→****н****,* ***r****→****р****,* ***ș****→****ш****,* ***t****→****т****,* ***ț****→****ц****,* ***v****→****в****,* ***z****→****з*** *(****bardă - бардэ****,* ***zarvă - зарвэ****,* ***măr - мэр****).*
* Reguli de transcriere pentru literele ***î*** și ***â***. Îmbinările ***âi*** sau ***îi*** se vor transcrie în ***ы*** pentru cuvintele ***mâine, pâine, câine*** și derivatele lor (***mâine - мыне, pâine - пыне, câine - кыне, mîine - мыне, pîine - пыне, cîine - кыне***). În alte situații se va aplica regula „literă → literă”: ***â****→****ы, î****→****ы*** (***român*** *→* ***ромын, întâi*** *→* ***ынтый***).
* Reguli pentru **ea** și **ia**. Se transcriu în **я**, cu o singură excepție: pronumele **ea** se transcrie **еа**; în același timp, verbul **ia** se transcrie ca **я**;
* Transcrierea lui **i** are loc prin trei litere diferite: **и**, **й**, **ь**. Concomitent, menționez existența cazurilor cînd litera **i** este omisă (***iepure*** *→* ***епуре***), sau trecută în litera ***ы*** (***introducere*** *→* ***ынтродучере***).

Reguli de transcriere pentru litera ***c***:

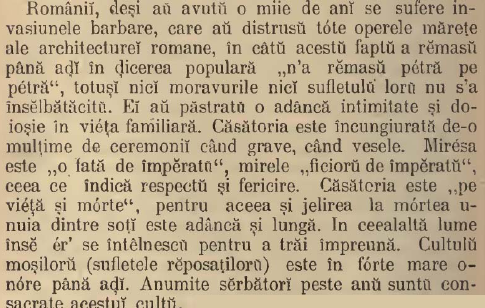
* ***c→к***, dacă după ***c*** urmează una din vocalele ***a***, ***â***, ***î***, ***o***, ***u*** sau o consoană diferită de ***h*** (***încrețit → ынкрецит, clocot → клокот, casă → касэ, cucoș → кукош, câmp → кымп***).
* combinațiile ***che***, ***chi*** se vor transcrie în ***ке*** și, respectiv ***ки*** (***cheltuială → келтуялэ, chihlimbar → кихлимбар, chibzui → кибзуи***).
* Dacă după îmbinarea ***ce*** nu urmează litera ***a***, atunci se aplică regula ***ce→че*** (***cercel → черчел, cep → чеп***).
* ***cea→ча*** (***ceară → чарэ, ceas → час, ceață → чацэ, ceașcă → чашкэ***). Excepție pentru articolul demonstrativ ***cea*** (***acea***) ***→*** ***чя*** (***ачя***).
* Dacă după îmbinarea ***ci*** nu urmează una din vocalele a, o, u, atunci se aplică regula ***ci→чи*** (***ciment → чимент, ciclu → чиклу, cimbrișor → чимбришор***). Dacă cuvîntul se termină în ***ci***, atunci poate fi aplicată una din regulile: ca excepție ***ci→ч*** (a**rici *→* арич, beci *→*беч, prichici *→* прикич**); ***ci - чь*** pentru plural (***saci - сачь, maci - мачь***); ***ci→чи*** pentru alte situații (***aci → ачи, răci → рэчи, înveșnici → ынвешничи***).
* ***cio→чо*** (***ciorbă → чорбэ, ciocârlie → чокырлие, cioban → чобан, cocioabă → кочоабэ***).
* ***ciu→чу*** (***ciuperci → чуперчь, ciubotă → чуботэ, bucium → бучум***).

Utilizând astfel de reguli (lista cărora poate fi prelungită), procesul de transliterare se transformă într-o acțiune de trecere prin „ciur și prin dârmon”. Pornind de la lexiconul contemporan al limbii române se stabilește un set de filtre, fiecare filtru având un coeficient de prioritate, care depinde de probabilitatea obținerii unui rezultat corect la aplicarea regulilor acestui filtru. Mai întâi, se vor aplica acele filtre care exclud, sau minimizează, intervenția manuală. Cuvintele filtrate se exclud din lexicon și asupra lexiconului rămas se aplică alte filtre. Din păcate. toate aceste etape de filtrare necesită un anumit grad de intervenție manuală.

**1.11 Procesarea textelor tipărite cu alfabet latin și litere adiționale**

Pentru ilustrarea tehnologiei descrise vom cerceta procesul de recunoaștere și verificare a unui text digitizat din cartea, tipărită în anul 1894 (fig. 10). Textul din figura 10 a fost recunoscut cu sistemul OCR IRIS. Ca urmare au rămas nerecunoscute cuvintele ortografiate cu litere specifice secolului al XIX-lea. De exemplu, se obține **tnsălbătăcitu** în loc se **însélbătăcitu**.

Acest rezultat nu poate fi îmbunătățit, deoarece IRIS nu posedă capacitatea de a selecta fragmente arbitrare din imagine. Utilizarea unui lexicon modern permite să se recunoască **avutú** ca **avută**, varianta corectă pentru acest context fiind **avut**. Cuvintele specifice lexiconului secolului al XIX-lea nu pot fi recunoscute corect, deoarece pentru aceasta sunt necesare dicționare corespunzătoare perioadei dare care, în cazul nostru, ar conține cuvintele **rămasú, viéța, împératú** etc.



**Figura 10**. Text digitizat, 1894 (Densuşianu, 1984, p. 130)

Mai jos, în fig. 11 este reprezentat textul din de mai sus, recunoscut cu ajutorul sistemului OCR IRIS. Este vizibilă rata majoră a erorii al acestui sistem de recunoaștere.

|  |
| --- |
| Româniî, deşi au avută o miie de an! se sufere in ­vasiunele barbare, care au distrusă tote operele măreţe ale architectureT romane, în cătu acesta faptă a rămasă până adt în dicerea populară "n 'a rămasă petră pe petră", totuşt nici moravuri1e niel sufletuln loru nu s'a tnsălbătăcitu. Ei au păstrate o adâncă intimitate şi doioşie in vieţa familiară. Căsătoria este încungiurată de-o mulţime de ceremonil când grave, când vesele. Miresa este "o, fată de tmpăratn", mirele "ficioru de împărată", ceea ce indică respectă şi fericire. Căsătoria este "pe vieţă şi morte", pentru aceea şi jelirea la m6rtea u­nuia dintre sot} este adâncă şi lungă. In ceealaltă lume tnsă er' se tntelnescn pentru a trăi împreună. Cultulă moşiloră (sufletele răposatilorn) este în forte mare 0- n6re până adt. Anumite sărbătort peste anu suntu con­sacrate acestui cultă. Figura. 11. Textul din fig. 10 recunoscut cu ajutorul sistemului OCR IRIS |

Dacă în textul recunoscut se vor restabili literele specifice și textul obținut se va verifica cu ajutorul corectorului ortografic, care posedă un lexicon al limbii române moderne de circa un milion de cuvinte, vom constata că 57 la sută din cuvintele textului sunt recunoscute drept corecte. Acestea sunt cuvintele, ortografia cărora a rămas intactă față de perioada secolului al XIX-lea, de exemplu **sufere, aceasta, fericire**. Cuvintele „suspicioase” sunt cele afectate de modificări ortografice, de exemplu **ceealaltă** (***cealaltă***), **doioșie** (***duioșie***), **miie** (***mie***), **avutú** (***avut***), **aɖĭ** (***azi***).

Pentru recunoașterea corectă a textului trebuie de instruit sistemul OCR ca să recunoască literele și să completeze lexiconul cu cuvinte noi, specifice secolului al XIX-lea. De exemplu: **avutú, o miie, invasiunele, pétră, nicĭ, sufletulú, lorú, însălbătăcitú, doioșie, viéța, ficiorú, împăratú, miresa** etc. Ținând cont că sistemul OCR ABBYY FineReader este înzestrat cu facilități de instruire, am mai efectuat un experiment. Sistemul a fost instruit în mod special ca să poată recunoaște literele specifice secolului al XIX-lea. Mai jos prezint câteva din literele care au fost utilizate în procesul de instruire:

* **ü** (literă finală, mută sau citită),
* **é** (é se pronunță ca diftongul ea),
* **ó** (ó se pronunță ca diftongul oa),
* **ɖ** (ɖ se citea ca z sau dz, în dependență de conținut),
* **ê** (ê se folosea ca litera â).

Unele rezultate ale experimentelor sunt relatate în Tabelul 2. Pentru a obține rezultate mai performante la verificarea textelor tipărite este necesar ca pentru perioada istorică corespunzătoare:

* să fie instruit scanerul pentru a recunoaște caracterele specifice;
* să fie elaborat un lexicon cu cuvinte și fraze uzuale specifice perioadei;
* să fie extinse facilitățile corectorului ortografic (spellchecker) pentru a utiliza și lexiconul elaborat.

**1.12 Procesarea textelor tipărite cu alfabete tranziționale sau cel slavon (sec. XVIII - XIX)**

Există cel puțin șapte versiuni ale alfabetului tranzițional (mixt chirilic-latin). Majoritatea literelor acestor alfabete pot fi recunoscute de ABBYY FineReader prin evidențierea codurilor respective din setul Unicode. O singură literă specifică pentru aceste alfabete lipsește în unicode - Ѡ. În acest caz urmează să fie inclusă o variantă de literă echivalentă (de exemplu o săgeată ↑ sau litera slavonă „yus” Ѧ ѧ) și instruit sistemul pentru recunoașterea acestei variante grafice.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tabelul 3:** Rezultatele experimentelor OCR cu texte din secolul XIX-lea | | |  |
| **Modul de recunoaștere** | **Cuvinte corecte** | **Cuvinte suspecte** | |
| IRIS | 57% | 43% | |
| ABBYY FineReader, fără nici o instruire | 63% | 37% | |
| ABBYY FineReader, cu instruire și dicționar pentru o pagină | 98% | 2% | |
| ABBYY FineReader, cu instruire, mai multe pagini, aceiași carte | 95% | 5% | |
| ABBYY FineReader, cu instruire, pagini din altă carte | 95.4% | 4.6% | |

În figura 12 este prezentat un text în grafie slavonă din anul 1786, iar în figura 13 același text recunoscut cu ajutorul sistemului OCR ABBEYY FineReader. Evident, FineReader cu instruire și dicționar, astfel sistemul OCR dat dispune de un pattern care minimizează rata erorii la recunoaștere.



**Figura 12**. Text slavon tipărit în anul 1786.

În figura 13 este prezentat textul de mai sus recunoscut cu ajutorul ABBEYY FineReader antrenat și cu dicționar de referință.

|  |
| --- |
| НОИ ЇѠСИФꙊ АЛДОИЛѢ  КꙊ МИЛА ЛꙊИ ДꙦНЕȝЇЕꙊ АЛЕСꙊ  ꙟпъρатꙋл ρоманилоρ пꙋρꙋρѣ авгꙋстꙋ, Кρаюл апостолическꙋ Геρманїеи, Ꙋнгаρїеи, Боемїеи, Далмацїеи, ши Склавонїеи, Аρχидꙋѯꙋл Аꙋстρїеи, Дꙋѯꙋл Бꙋρгꙋндїеи, ши Лотаρиингїеи, Маρе Дꙋѯꙋл Етρꙋρїеи, Маρе Пρинципꙋл Аρдѣлꙋлꙋи, Дꙋѯꙋл Медїѡланꙋлꙋи, Мантовеи, Паρмеи ши чѣ Комитꙋл χабспꙋρгꙋлꙋи, Фландρїеи, ши атиρолꙋлꙋи.  Щїꙋтꙋ ши фъρъ ꙟдоѧлъ лꙋкρꙋ есте, кꙋмкъ спρе феρичиρѣ де ѡбще фоаρте мꙋлтꙋ фаче ꙋн дρептꙋ ашеȝъмѫнтꙋ де даρе (саꙋпоρцїъ) пρин каρе гρъꙋтъциле чѣле де ѡбще, а ꙟпаρтꙋ дꙋпъ ѡ адевъρатъ потρивρе (пρопоρцїе), ѩρ ꙟпρотивъ, ѡ недρѣптъ ρѫндꙋѧлъ де даρе нъдꙋшѣще кꙋлтꙋρа ши сѫρгꙋинца (индꙋстρїа) ши ꙟмꙋлциρѣ де ноρодꙋ (попꙋлацїа) |

**Figura 13**. Textul din fig. 12 recunoscut cu ajutorul sistemului OCR FineReader.

Pentru lexiconul original românesc procesul respectiv poate fi parțial automatizat. În acest scop s-au stabilit reguli de transcriere a literelor și îmbinărilor de litere. Exemplu de aceste reguli sunt prezentate mai jos:

* Reguli de transcriere „literă → literă”. De exemplu, а*→****a****,*в*→****v****,*д*→****d****,*ρ*→****r****,*ꙋ*→****u****,*о*→****o****,*ш*→****ș****,*и*→****i****,*с*→****s****,*χ*→****h****,*к*→****c****,*ї*→****i****,*м*→****m****,*н*→****n*** *(*лꙋкρꙋ ***- lucru****,* пρопоρцїе ***- proporție****,*маρе ***- mare****).*
* Reguli de transcriere pentru litera ꙟ. Caracterul ꙟ se transcrie ca ***îm*** sau ***în*** dacă se află la începutul cuvântului, și va depinde de litera nemijlocită ce va urma după ꙟ. Dacă următoarea literă este п sau б atunci se aplică regula ꙟ*→* ***îm*** (ꙟпъρат*→* ***împărat***), în caz contrar se va aplica regula ꙟ*→* ***în*** (ꙟдоѧлъ*→* ***îndoială***). În caz că următoarea literă este

м sau н se va aplica regula „literă → literă”: ꙟ*→* ***î*** (ꙟнсъ*→* ***însă***).

* Reguli de transcriere pentru litera ѧ. Caracterul ѧ se transcrie ca ***ia*** sau ***ea*** dacă nu se află la începutul cuvântului și litera predecesoare nu a fost ї,е, sau и (їѧρ *→* ***iar***). În acest caz pentru litera dată se vor aplica regulile standarde ortografice de trecere în ***ia*** sau ***ea***, în dependență de context (ꙟдоѧлъ *→*  ***îndoială***).
* Transcrierea literei ъ poate fi transcrisă prin două litere diferite: **ă**, **e**. Dacă se află la sfârșitul cuvântului, iar litera precedentă a fost o consoană atunci se aplică regula ъ*→****ă*** (фъρъ*→* ***fără***, цаρъ*→* ***țară***). În cazul repetării conjuncției съ съ (***să să***) se va aplica întâi legea ъ*→****ă*** iar apoi ъ*→****e***  (съ съ *→* ***să se***).
* Pentru litera щ se vor aplica următoarele legi: щ*→****ș*** dacă următoarea literă este т, sau щ*→****șt*** în caz contrar (ѡбще *→* ***obște***).
* Litera ѩ se va transcrie conform regulii ѩ*→****ia*** (ѩρмаρок*→****iarmaroc***) dacă litera precedentă nu este ї sau и. În caz contrar se aplică regula ѩ*→****a***.

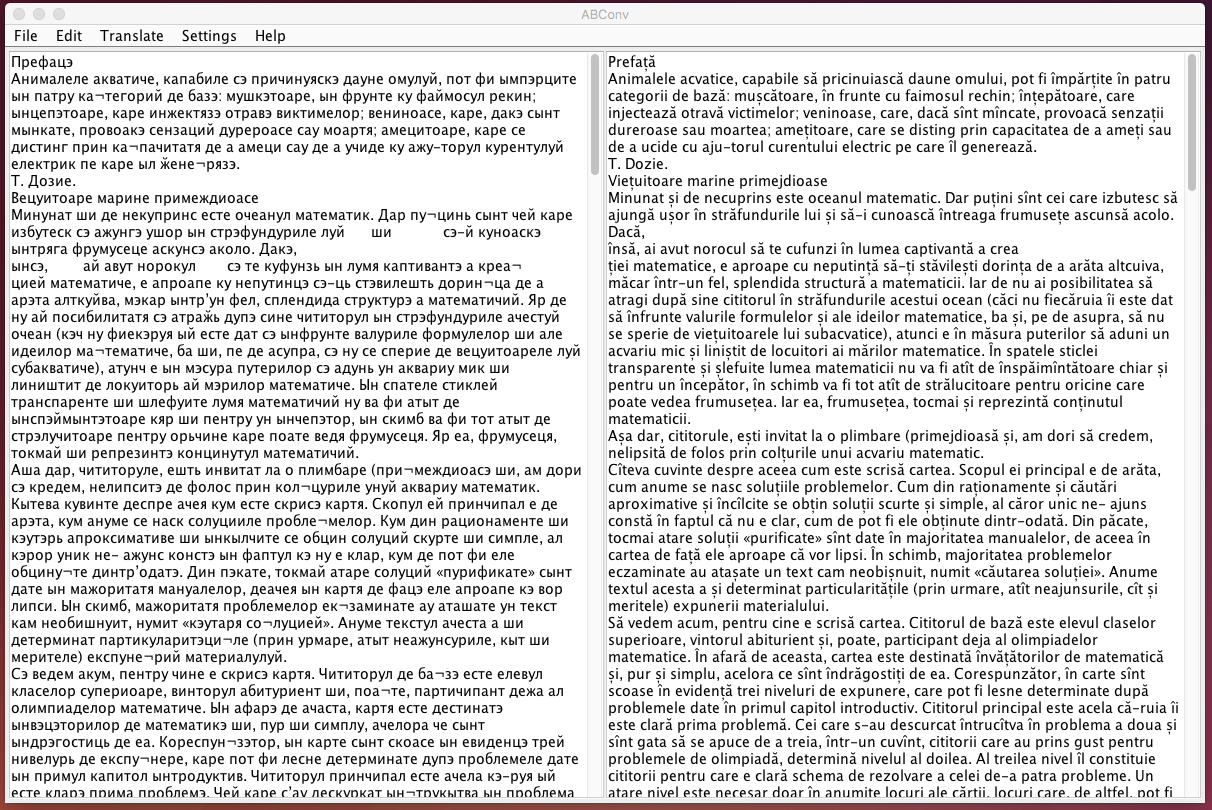
**2. APLICAȚIA PRACTICĂ, ARHITECTURA ŞI DEZVOLTAREA PROIECTULUI**

**2.****1 Arhitectura aplicației**

Scopul primar al aplicației realizate a fost implementarea algoritmului de convertare de la grafia chirilică la cea latină descris anterior, într-un mod eficient. Deci sa propus ca rata erorii al aplicației date să nu depășească 0,01%, astfel numărul de cuvinte ce au fost transliterate cu greșeli (fie ortografice sau lexicografice) să nu depășească 0,01% din numărul total de cuvinte convertate. Asigurarea unui astfel de procentaj de corectitudine depinde în cea mai mare parte de însăși logica aplicației (deci algoritmului proiectat). Totodată, sa propus și elaborarea unei interfețe grafice cât mai intuitivă și ușor de perceput.

În general, fiecare aplicație indiferent de tipul ei trebuie să fie unică, deci să aibă elemente distincte atât la nivel de design cât și la nivel tehnic, ceea ce ține de funcționalitățile oferite utilizatorului. Programatorul are datoria de a realiza o aplicație unică prin designul interfeței grafice și funcționalitățile utilizate. O aplicație bine realizată va fi ușor accesibilă atât la nivel vizual cât și tehnic, iar conținutul prezentat și specificul acestuia trebuie să fie conceput astfel încât utilizatorului să-i fie cât mai ușor să-l acceseze.

Realizarea oricărui proiect începe cu planificarea, și în etapa de planificare trebuie stabilite tipul de aplicație(instituțională, educativă, de prezentare a unui produs etc.), ce instrumente vor fi oferite utilizatorului, preferințele vizuale, structura aplicației la lansare, ce informații vor fi la prima vedere, unde și cum vor fi amplasate butoanele, meniurile, tehnologiile necesare pentru realizarea aplicației, etc. Toate aceste informații sunt foarte importante în realizarea interfeței, deoarece influențează foarte mult și în mod direct forma finală a aplicației proiectate. Interfața grafică este prima ce utilizatorul vede și unica lui sursă de a comunica cu algoritmul aplicației, respectiv aspectul ei vizual și comoditatea utilizării este unul din pilonii principali pe care se bazează dezvoltarea unei aplicații. O aplicație bogată, cu structură de meniuri mare și multe informații nu poate fi în principiu foarte intensă la nivel vizual, deoarece în acest mod va fi subminat scopul aplicației prin distragerea atenției de la conținut, și respectiv reducerea importanței sale. Astfel, o interfață trebuie să fie mai simplă, mai concisă, pentru a pune în valoare conținutul și instrumentele necesare, astfel a le expune utilizatorului într-un mod simplu și accesibil.

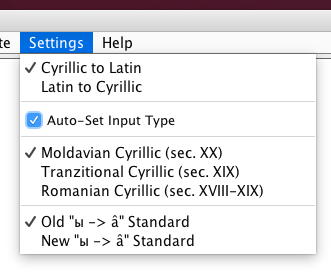


**Figura 15.** Aplicația ABConv, cu exemplu de text chirilic sovietic transliterat.

Aplicația permite transliterarea a textelor chirilice din trei perioade de timp diferite:

* texte chirilice tipărite pe spațiul RSS Moldovenești de la mijlocul sec. XX până în 1989;
* texte chirilice tipărite cu alfabet slavon (sec. XVIII - XIX), ca de exemplu manuscrise vechi bisericești;
* texte chirilice tranziționale (sec. XIX - XX) care conțin atât caractere slavone vechi cât și caracterele alfabetului latin.

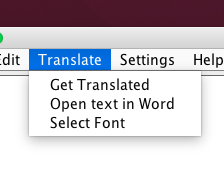
Au fost implementate toate trei algoritme de transliterare bazate pe regulile deduse în cap. 1.10 – 1.12. Utilizatorul poate alege manual ce tip de text (din ce perioadă istorică face parte) dorește să translitereze, utilizând meniul ***Settings*** din bara de meniu.



**Figura 16**. Compartimentul meniului *Settings* al aplicației ABConv

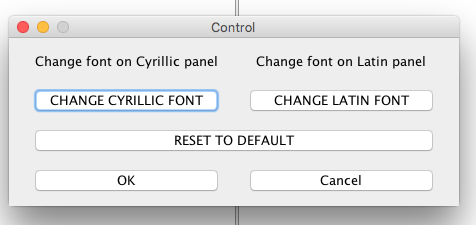
După cum se vede în screenshot-ul meniului, utilizatorul poate alege următoarele setări ale aplicației:

* „***Cyrillic to Latin***” sau „***Latin to Cyrillic***” – direcția generală a transliterării, din chirilică în latină sau invers. Aplicația permite transliterarea din latină în chirilică, doar cu unele mici dezavantaje: așa cum „chirilizarea” nu a fost scopul principal la realizarea aplicației date, algoritmul de transliterare din grafia latină în cea chirilică nu a fost încă perfecționat. Așadar rata erorii la transliterarea din latină în chirilică este semnificabil de majoră. Pe de altă parte, rata erorii la transliterarea inversă – din chirilică în latină – este nesemnificativă, nu întrece 0.01%;
* „***Auto-Set Input Type***” – funția dată procesează automat textul introdus de către utilizator și decide din ce perioadă istorică face parte, transliterând-ul conform algoritmului necesar. Așadar utilizatorul nu trebuie numaidecât să știe de algoritm să selecteze, funcția dată poate decide automat, judecând după conținutul textului introdus. Evident, utilizatorul poate porni sau opri funcția dată bifând sau nu checkbox-ul respectiv (by default el este bifat);
* „***Moldavian Cyrillic (sec. XX)***”, „***Tranzitional Cyrillic (sec. XIX)***” sau „***Romanian Cyrillic (sec. XVIII–XIX)***” – reprezintă cele trei perioade istorice din care pot face parte textele chirilice spre transliterare. Utilizatorul poate selecta oricare din aceste trei perioade și apoi introduce textul său dorit, sau poate bifa celula de mai sus (Auto-Set Input Type) și aplicația singură va decide ce fel de text a fost introdus și îl va translitera conform algoritmului necesar;
* „***Old “ы -> â” Standard*** ” sau „***New “ы -> â” Standard***” – reprezintă cele două standarde de scriere a literei ***â***. Conform legilor noi introduse în DEX litera „***â***” urmează să fie scrisă nu doar în cuvintele ce au rădăcina „rom” (precum ***românie, aromân***) ci în toate cuvintele ce până acum erau scrise cu litera „***î***” cu excepția celor ce încep sau sfârșesc cu litera „***î***”, sau conțin un prefix înainte de ea (precum ***neînțeles***). Așadar utilizatorul are posibilitatea să aleagă ce standard dorește să respecte algoritmul de transliterare cel mai nou, descris anterior, sau așa cum a fost până la introducerea standardului nou (***când*** – ***cînd***).



**Figura 17**. Compartimentul meniului *Translate* al aplicației ABConv

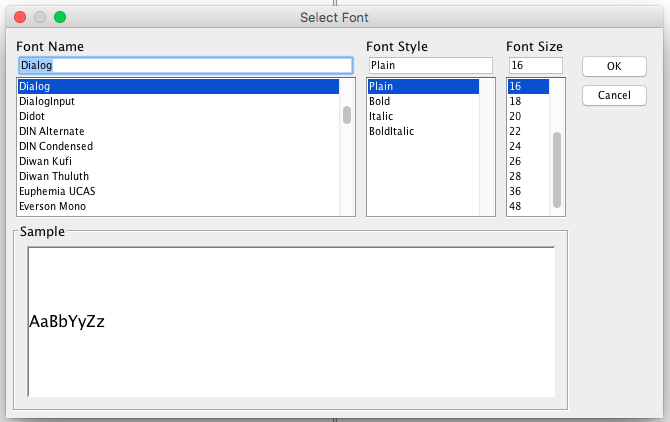
Accesând meniul ***Translate*** utilizatorul are la îndemână următoarele funcții:

* „***Get Translated***” – transliterează textul de pe panoul din stânga (panoul chirilic) conform algoritmului selectat de către utilizator (selectarea are loc în compartimentul ***Settings***). Așadar utilizatorul poate manual să introducă text de la tastieră sau să facă *paste* la un text chirilic pe panoul din stânga al aplicației, să dea click pe butonul dat și va obține pe panoul din dreapta textul respectiv transliterat;
* „***Open Text in Word***” – funcția dată salvează într-un fișier textul transliterat (de pe panoul din dreapta) și îl deschide cu MS Office Word, dacă acesta este instalat în calculatorul utilizatorului. Funcția dată permite utilizatorului într-un click să obțină textul transliterat în Office Word pentru a-l putea controla cu spellcheck-erul la greșelile ortografice care ar putea rezulta în urma transliterării;
* „***Select Font***” – funcția dată permite utilizatorului să selecteze font-ul dorit pentru fiecare din panelele de text (pentru cel cu text chirilic și cel cu text latin). Apăsând pe butonul dat se va deschide fereastra din Figura 18.

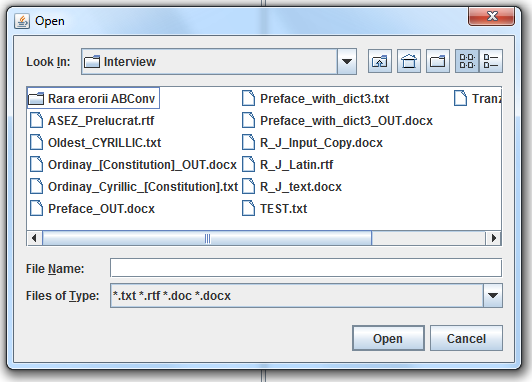
**Figura 18**. Submeniul „*Select Font*”

Așadar, utilizatorul apăsând pe oricare din butoanele „*CHANGE CYRILLIC FONT*” sau „*CHANGE LATIN FONT*” va accesa fereastra ce-i va permite să selecteze fontul, stilul și mărimea dorită. Totodată este prezentă și o previzualizare a fontului selectat (Figura 19).

Fonturile selectate se voi păstra în fișierul config al aplicației, așadar la următoarea pornire a aplicației ele se vor păstra așa cum au fost selectat. Dacă utilizatorul dorește să reîntoarcă setările inițiale ale aplicației – atunci este de ajuns să dea click pe butonul „*RESET TO DEFAULT*”.

**Figura 19**. Fereastra „Select Font” aplicației ABConv

Aplicația ABConv permite introducerea fișierelor cu text prin intermediul ferestrei standarde

JFileChooser, cât și prin „aruncarea” fișierelor direct pe interfața grafică a aplicației. Deci, cel mai recent, am implementat funcția Drag’n Drop, care ușurază considerabil lucrul utilizatorului cu utilajul software dat. Aplicația permite citirea fișierelor de format text cu extensia **.DOCX, .DOC, .RTF .TXT**. Toate fișierele cu altă extensie nu sunt acceptate la filtrate, și deci nu vor fi procesate.

**Figura 20**. Meniul standard de selectare a fișierelor de intrare

**2.2 Tehnologii utilizate**

Pentru realizarea GUI-ului a fost utilizată biblioteca de dezvoltare pentru interfețe grafice ***Java Swing***, care conține mai multe componente. Toate obiectele Swing, cu excepția clasei **JFrame**, moștenesc clasa *javax.swing.JComponent*, care la rândul ei moștenește (indirect) clasa *java.awt.Container*. Astfel, se creează o ierarhie de componente, fiecare element (numit *parent*) înglobând alte sub-componente (numite *children*). Componentele utilizate:

* *javax.swing.JPanel*

**JPanel** este un container generic care poate conține alte elemente. Poate fi vizibil, schimbându-i-se culoarea background-ului, sau modelul marginii, sau invizibil, folosit doar pentru ierarhizarea conținutului. Este indicat să nu plasați alte obiecte direct pe un *JFrame*, ci doar un JPanel care să conțină restul de elemente. Acest lucru este foarte util și când avem nevoie să schimbăm complet elementele unui JFrame, înlocuim doar JPanel-ul;

* *javax.swing.JLabel*

**JLabel** este un component utilizat pentru a afișa text sau imagini într-un container;

* *javax.swing.JButton*

**JButton** este, cum îi spune și numele, un buton. Acesta poate avea afișat un text sau o imagine. De cele mai multe ori este utilizat asociindu-i-se un Event Handler de tip ActionListener care se declanșează când acesta este apăsat (se dă click pe el);

* *javax.swing.JTextField*

**JTextField** reprezintă o zonă în care se poate introduce text scurt, de o singură linie. Și acestul element i se poate asocia un Event Handler de tip ActionListener care se declanșează când se apasă tasta Enter în zona de editare;

* *javax.swing.JTextArea*

**JTextArea** reprezintă o zonă în care se poate introduce text de mai multe linii.

În antetul clasei care răspunde de crearea interfeței grafice a fost necesar de importat toate aceste librării.

**import** **javax.swing.BorderFactory**;

**import** **javax.swing.JCheckBox**;

**import** **javax.swing.JFileChooser**;

**import** **javax.swing.JFrame**;

**import** **javax.swing.JMenu**;

**import** **javax.swing.JMenuBar**;

**import** **javax.swing.JMenuItem**;

**import** **javax.swing.JPanel**;

**import** **javax.swing.JScrollPane**;

**import** **javax.swing.JSeparator**;

**import** **javax.swing.JTextPane**;

**import** **javax.swing.filechooser.FileFilter**;

**import** **javax.swing.JOptionPane**;

**import** **java.awt.BorderLayout**;

**import** **java.awt.Desktop**;

**import** **java.awt.Dimension**;

**import** **java.awt.GraphicsEnvironment**;

**import** **java.awt.GridLayout**;

**import** **java.awt.event.ActionEvent**;

**import** **java.awt.event.ActionListener**;

**import** **java.io.BufferedReader**;

**import** **java.io.FileWriter**;

**import** **java.io.IOException**;

**import** **java.io.InputStreamReader**;

Iar apoi de creat fereastra principală a aplicației, setând mărimea default a JFrame-ului (în cazul dat fiind 1200 pe 800 pixeli). Tot în metoda dată am setat componentele barei de meniu și fiecare ActionListener ce va corespunde butoanelor respective.

JFrame theFrame = **new** JFrame("ABConv"); theFrame.setDefaultCloseOperation(JFrame.DO\_NOTHING\_ON\_CLOSE);

BorderLayout layout = **new** BorderLayout();

JPanel background = **new** JPanel(layout);

background.setBorder(BorderFactory.createEmptyBorder(4, 4, 4, 4));

JMenuBar menuBar = **new** JMenuBar();

JMenu fileMenu = **new** JMenu("File");

JMenu editMenu = **new** JMenu("Edit");

JMenu helpMenu = **new** JMenu("Help");

JMenu translateMenu = **new** JMenu("Translate");

JMenu standardMenu = **new** JMenu("Settings");

theFrame.setJMenuBar(menuBar);

JTextPane cyrillic = **new** JTextPane();

JTextPane latin = **new** JTextPane();

JScrollPane scrollPaneC = **new** JScrollPane(cyrillic);

JScrollPane scrollPaneL = **new** JScrollPane(latin);

GridLayout grid = **new** GridLayout(1, 2);

grid.setHgap(2);

JPanel mainPanel = **new** JPanel(grid);

mainPanel.add(scrollPaneC);

mainPanel.add(scrollPaneL);

background.add(BorderLayout.CENTER, mainPanel);

theFrame.getContentPane().add(background);

theFrame.pack();

theFrame.setSize(1200, 800);

theFrame.setVisible(**true**);

**2.3 Descrierea algoritmelor, regulilor și excepțiilor**

Mai jos este elaborat codul algoritmului ce determină tipul textului introdus de către utilizator (cu condiția că este bifată căsuța „***Auto-Set Input Type***”).

**try** {

**char**[] inputChars = inputText.toCharArray();

**boolean** oldCyr = **false**, tranzCyr = **false**;

**int** nmax = 550; *//nr maxim de caractere procesate*

**for** (**int** i = 0; i < inputChars.length; i++) {

**if** (inputChars[i] == '\ua64a' || inputChars[i] == '\u021d' ||

inputChars[i] == '\ua65f' || inputChars[i] == '\ua64b' ||

inputChars[i] == '\u0457' || inputChars[i] == '\u046f' ||

inputChars[i] == '\u046b')

{

oldCyr = **true**;

**break**;

}

**if**(i>nmax) **break**;

}

**if**(oldCyr){

**for** (**int** i = 0; i < inputChars.length; i++) {

**if**(inputChars[i] == '\u012d' || inputChars[i] == '\u0069' ||

inputChars[i] == '\u006e' || inputChars[i] == '\u007a' ||

inputChars[i] == '\u006d' || inputChars[i] == '\u0445' ||

inputChars[i] == '\u0064' || inputChars[i] == '\u0440')

{

tranzCyr = **true**;

**break**;

}

**if**(i>(nmax+100)) **break**;

}

}

**if**(tranzCyr){

ActionEvent ev = **null**;

TranzitCyrillicInput tranz = **new** TranzitCyrillicInput();

tranz.actionPerformed(ev);

}**else** **if** (oldCyr){

ActionEvent ev = **null**;

OldestCyrillycInput oldest = **new** OldestCyrillycInput();

oldest.actionPerformed(ev);

}**else**{

ActionEvent ev = **null**;

OrdinaryCyrillicInput ordinary = **new** OrdinaryCyrillicInput();

ordinary.actionPerformed(ev);

}

} **catch** (Exception e) {

infoBox(e.toString(), "AutoSet Input Type Error");

e.printStackTrace(System.out);

}

Din cele de mai sus este clar că metoda primește ca parametru un String cu textul introdus de către utilizator. Pentru a controla fiecare caracter în parte al String-ului dat a fost necesar să-l transform într-un array de caractere și printr-un simplu ciclu *for* să controlez dacă sunt prezente caracterele specifice textelor vechi slavone sau nu. Dacă persistă măcar un caracter specific acestor texte variabila booleană ***oldCyr*** primește valoarea **true** și ciclul *for* se stopează. Apoi dacă variabila ***oldCyr*** este **true,** se controlează încă odată dacă textul nu conține caractere specifice alfabetelor tranziționale. Dacă da, atunci variabila booleană ***tranzCyr*** primește valoarea **true** și ciclul *for* se stopează. Apoi se controlează, dacă variabila ***tranzCyr*** este **true** atunci textul este tranzițional, dacă variabila ***oldCyr*** este **true** atunci textul este vechi slavon, iar dacă nici una din aceste variabile nu este **true**, atunci textul este simplu chirilic (sovietic) din perioada sec. XX.

Metoda controlează maxim 550 de caractere, ceea ce este mai mult de ajuns pentru a determina tipul textului introdus. Mărirea numărului de caractere va mări și utilizarea CPU la fiecare procesare a unui nou document, și timpul execuției va dura mai mult.

Mai jos sunt prezentate primele rânduri din codul algoritmului de transliterare a textelor vechi slavone:

**for** (**int** i = 0; i < incStr.length(); i++) {

tPrevChar = (i > 2) ? incStr.charAt(i - 3) : '^';

sPrevChar = (i > 1) ? incStr.charAt(i - 2) : '^';

prevChar = (i > 0) ? incStr.charAt(i - 1) : '^';

currentChar = incStr.charAt(i);

nextChar = (i < incStr.length() - 1) ? incStr.charAt(i + 1) : '$';

sNextChar = (i < incStr.length() - 2) ? incStr.charAt(i + 2) : '$';

tNextChar = (i < incStr.length() - 3) ? incStr.charAt(i + 3) : '$';

**char** prevTr;

String S1= "абвдезилмноупрстэ\ua64bфхцш\u012CЪАБВДЕЗИЛМНОПРСТ\uA64AФНЦ";

String S2= "abvdezilmnouprstăufhțş\u012CĂABVDEZILMNOPRSTUFHȚ";

String S3;

String S4;

**int** p = S1.indexOf(currentChar);

**if** (p != -1) {

out.append(S2.charAt(p));

} **else** {

**switch** (currentChar){

**case** '\ua666':

out.append('M');

**break**;

**case** '\u021d':

**if** (Character.isUpperCase(nextChar) ||

Character.isUpperCase(prevChar))

{

out.append('Z');

}**else**{

out.append('z');

}

**break**;

**case** '\u0457':

out.append('i');

**break**;

**case** '\u0407': *// Ї*

out.append('I');

**break**;

**case** 'к':

S3 = "\u0457еиюяЕИЮЯ";

S4 = "сзСЗ";

**if** (S3.indexOf(nextChar) != -1) {

out.append('c');

out.append('h');

**break**;

}

**if** (S4.indexOf(nextChar) != -1) {

out.append('x');

i++;

**break**;

}

out.append('c');

**break**;

**case** '\ua65f':

S3 = "рпРП";

**if** (S3.indexOf(nextChar) != -1) {

out.append('î');

out.append('m');

**break**;

}

S4 = "левмтсдкцЛЕВМДТСКЦ";

**if**(prevChar == '\u0020'){

**if**( (S4.indexOf(nextChar) != -1) ||

(nextChar == '\u0020') )

{

out.append('î');

out.append('n');

**break**;

}

}

out.append('î');

**break**;

Mai sus sunt prezentate doat câteva din toate *case*-urile srise pentru procesarea întregului alfabet slavon, astfel încât algoritmul să țină cont de toate excepțiile și cazurile speciale în transliterarea textelor slavone vechi. Algoritmul nu face altceva decât procesează fiecare caracter din textul introdus (stocat în StringBuilder-ul *incStr*) și îi corespunde fiecărui caracter procesat unul respectiv în StringBuilder-ul de ieșire *out*. Pentru cazurile simple (precum literele ***a, b, e, t*** etc.) se folosește secvența:

**int** p = S1.indexOf(currentChar);

**if** (p != -1) {

out.append(S2.charAt(p));

}

Iar pentru cazurile speciale sunt scrise câte un *case* aparte. Din câte se vede caracterele au fost specificare în formatul lo Unicode pentru a ușura lucrul, deoarece nici IDE-ul utilizat pentru dezvoltarea aplicației nu poate reprezenta corect caracterele Unicode precum **\ua65f** sau **\u0407** (în locul lor sunt reprezentate glife goale).

**2.4 Exemplu practic de text transliterat**

În **Figura 21** este prezentat un text slavon inițial din anul 1786, digitizat și apoi recunoscut cu ajutorul sistemului ABBEYY FineReader. Iar în **Figura 22** este reprezentat același text slavon, doar că deja transliterat cu ajutorul aplicației ABConv.

|  |
| --- |
| Даρ нꙋ маи пꙋцинꙋ адевъρатꙋ, ши пρин испитиρе ꙟтъρитꙋ лꙋкρꙋ есте ши ачеста, кꙋмкъ даρѣ аρꙋнкатъ пънъ акꙋм пρин χотъρѫρѣ дїетїи (адꙋнъρїи де цаρъ) ши сꙋмма чѣ де песте анꙋ, каρѣ еρа ꙟвоитъ пентρꙋ ꙟдестꙋлаρѣ тρебꙋинцелоρ ши кїелтꙋелилоρ доместече, нꙋ нꙋмаи ꙟтρе ѡаменїи деѡсъби, чи ниче ꙟтρе ꙟселе комитатꙋρи нꙋ еρа ꙟпъρцитъ дꙋпъ пρавилиле адевъρатеи потρивиρи: фїинд къ ши ꙟтρѫнселе адꙋнъρїи де цаρъ, атѫтꙋ аша нꙋмителе поρте саꙋ поρцїи (песте тотꙋ гρъинд) волничѣще саꙋ ꙟпъρцитꙋ спρефїеще каρе комитатꙋ, кꙋмши аρꙋнкаρѣ ши χотъρѫρѣ дъρилоρ пρин дикацїе, дела каρѣ съ чеρѣ ши съ ρъдика даρѣ саꙋ фъкꙋтꙋ гρешитꙋ, недρептꙋ, ши асꙋпρитоρю индꙋстρїеи.  Дечи динтρачаста ѹρъмѣȝъ, кꙋмкъ адевъρата феρичиρе а цъρилоρ ꙋнгꙋρещи челоρ че сѫнтꙋ ꙟкρеȝꙋте пꙋρтъρїи ноастρе де гρижъ, ꙟтρачеста кип ниче ѡдинїѡаρъ нꙋ съ поате добѫнди, дꙋпъ ѹρмаρе датоρїа ноастρъ чѣρе, ка аша съ кибȝꙋим лꙋкρꙋл, кѫтꙋ тот сꙋпꙋсꙋл каρе есте сꙋптꙋ даρе, спρеачастъ гρеꙋтате де ѡбще съш дѣе паρтѣ са нꙋмаи дꙋпъ мъсꙋρа пꙋтеρїи сале, ши ꙟвоита спρе бꙋнꙋл де ѡбще депесте анꙋ сꙋммъ (кѫтꙋл) съ съ ꙟпаρтъ потρивитꙋ ꙟтρъ еи. Спρеачаста есте де тρебꙋинцъ, маи наинте де тоате а аρꙋнка ши а пꙋне даρе пе вρеꙋн лꙋкρꙋ (ѡбїектꙋ) ка ачела, ꙟтρꙋ каρе съ съ поатъ фаче ѡ адевъρатъ ꙟпъρциρе; адекъ съ съ пꙋе ши съ съ аρꙋнче даρе пе пъмѫнтꙋρи саꙋ фꙋндꙋρи, а къρоρа кꙋпρиндеρе ши ρодиρе съ съ ꙟсамне, ѩρ дин пρотивъ индꙋстρїа ла цаρъ пρиначаста съ съ мѫнтꙋѧскъ де тоатъ гρеꙋтатѣ ши асꙋпρиρѣ.  Кꙋм апои ва тρебꙋи съ съ испρъваскъ атѫтꙋ мъсꙋρаρѣ, кꙋмши χотъρѫρѣ тꙋтꙋρоρ ρодитоаρелоρ фꙋндꙋρи, ши ρодитоρїа ачелоρаш ꙟкѫтꙋ съ поате маи диρептꙋ, ши маи кꙋ ρѫндꙋ, аρатъ чѣ ла сфѫρшитꙋл ачещїа ѹρмътоаρе маи пелаρгъ ꙟвъцътꙋρъ.  Дечи тρебꙋе ꙟдатъ дꙋпъ ρѫндꙋитꙋл аколо кип съ съ апꙋче лꙋкρаρѣ, ши фїеще каρе тρебꙋе ла ачестꙋ спρе бꙋнꙋл де ѡбще слꙋжитоρю лꙋкρꙋ деачѣѧш маи таρе съ съ сѫρгꙋѧскъ ши съ гρъбѣскъ, кꙋ кѫтꙋ ши бꙋнꙋл де ѡсъби а фїеще къρꙋи дъждїашꙋ цъρанꙋ аρе ѡ фоаρте стρѫнсъ легътꙋρъ кꙋ дѫнсꙋл.  Ꙟ еъ пънъ акꙋм нꙋ нꙋмаи даρѣ цъρанилоρ еρа ашеȝатъ пе деȝпотρивите ши неадевъρате темеюρи, чи ши ρъдикаρѣ де ѡасте (инсꙋρекцїе) каρѣ дꙋпъ леџиле цъρїи тρебꙋѧ съ съ факъ де кътρъ боеρини (нѣмеши) ашиждеρъ съ ꙟтѫмпла ꙟтρꙋн кипꙋ несигꙋρꙋ, ши неловитоρю кꙋ мъсꙋρа адевъρатеи потρивиρи, ши фъρъде ниче ꙋн фолосꙋ кꙋρгътоρю спρе апъρаρѣ патρїеи. Дечи ши ачаста тρебꙋе съ съ мъсѡаρе маи диρептꙋ, ши маи кꙋ дѣмъρꙋнтꙋл, саꙋ ꙟлокꙋл ачеїа съ съ ρѫндꙋѧскъ ѡ мижлочиρе мъсꙋρатъ ши потρивитъ |

**Figura 21**. Textul slavon (an. 1786) recunoscut cu ajutorul sistemului FineReader.

|  |
| --- |
| Dar nu mai puținu adevăratu, şi prin ispitire întăritu lucru este şi acesta, cumcă darea aruncată pănă acum prin hotărîrea dietii (adunării de țară) şi summa cea de peste anu, carea era învoită pentru îndestularea trebuințelor şi chieltuelilor domestece, nu numai între oamenii deosăbi, ci nice între însele comitaturi nu era împărțită după pravilile adevăratei potriviri: fiind că şi întrînsele adunării de țară, atîtu aşa numitele porte sau porții (peste totu grăind) volniceaște sau împărțitu sprefiește care comitatu, cumşi aruncarea şi hotărîrea dărilor prin dicație, dela carea să cerea şi să rădica darea sau făcutu greşitu, nedreptu, şi asupritoriu industriei.  Deci dintraceasta urămează, cumcă adevărata fericire a țărilor ungurești celor ce sîntu încrezute purtării noastre de grijă, întracesta chip nice odinioară nu să poate dobîndi, după urmare datoria noastră ceare, ca aşa să chibzuim lucrul, cîtu tot supusul care este suptu dare, spreaceastă greutate de obște săş deae partea sa numai după măsura puterii sale, şi învoita spre bunul de obște depeste anu summă (cîtul) să să împartă potrivitu între ei. Spreaceasta este de trebuință, mai nainte de toate a arunca şi a pune dare pe vreun lucru (obiectu) ca acela, întru care să să poată face o adevărată împărțire; adecă să să pue şi să să arunce dare pe pămînturi sau funduri, a cărora cuprindere şi rodire să să însamne, iar din protivă industria la țară prinaceasta să să mîntuiască de toată greutatea şi asuprirea.  Cum apoi va trebui să să isprăvască atîtu măsurarea, cumşi hotărîrea tuturor roditoarelor funduri, şi roditoria aceloraş încîtu să poate mai direptu, şi mai cu rîndu, arată cea la sfîrşitul aceștia următoare mai pelargă învățătură.  Deci trebue îndată după rînduitul acolo chip să să apuce lucrarea, şi fiește care trebue la acestu spre bunul de obște slujitoriu lucru deaceaiaş mai tare să să sîrguiască şi să grăbească, cu cîtu şi bunul de osăbi a fiește cărui dăjdiaşu țăranu are o foarte strînsă legătură cu dînsul.  În eă pănă acum nu numai darea țăranilor era aşezată pe dezpotrivite şi neadevărate temeiuri, ci şi rădicarea de oaste (insurecție) carea după lejile țării trebuia să să facă de cătră boerini (neameşi) aşijderă să întîmpla întrun chipu nesiguru, şi nelovitoriu cu măsura adevăratei potriviri, şi fărăde nice un folosu curgătoriu spre apărarea patriei. Deci şi aceasta trebue să să măsoare mai direptu, şi mai cu deamăruntul, sau înlocul aceia să să rînduiască o mijlocire măsurată şi potrivită |

**Figura 22**. Textul slavon (an. 1786) recunoscut cu ajutorul sistemului FineReader.

**2.5 Împachetarea și distribuția**

├── Fonts/   
├── lib/

└── **DocDoxRead.jar**

└── **jrtf-0.7.jar**

└── **rtfparsekit.jar**

├── ABConv.jar

├── Dictionary.txt

├── Prefixes.txt

├── config.properties

La moment, aplicația ABConv constă din următoarele fișiere:

* ***ABConv.jar*** - fișierul executabil ce pornește aplicația de convertare;
* ***Dictionary.txt*** - dicționarul de cuvinte în care se stochează excepțiile algoritmului de convertare, în formatul "cuvînd chirilic"/"cuvînt latin", și au prioritate superioară în procesul convertării. Utilizatorul poate adăuga, modifica sau șterge cuvinte din dicționar direct din fereastra principală a aplicației;
* ***Prefixes.txt*** - fișierul în care se stochează o bază de date de prefixe al limbii române, care are aport la corectitudinea transliterării unor cuvinte (în special celor ce conțin litera "â");
* ***config.properties*** - fișierul în care se stochează configurațiile de sistemă ale aplicației, precum fontul ales sau mărimea lui;
* ***Folder-ul „Fonts”*** conține font-urile necesare pentru reprezentarea corectă al textelor ce conțin caractere slavone. La prima pornire a aplicației font-urile din mapa dată se vor instala automat dacă nu vor fi găsite prezente în mașina de calcul al utilizatorului;
* **Folder-ul „lib”** conține librăriile necesare pentru procesarea fișierelor **\*.docx \*.doc** și **\*.rtf**.

Acestea sunt fișierele ce se conțin în distributivul oficial al aplicației ABConv. Cu timpul aplicația va fi dezvoltată și componența fișierelor, evident, va fi modificată.

**CONCLUZII**

Scopul lucrării a constat în realizarea și implementarea unui sistem de transliterare a textelor din grafie chirilică/slavonă în cea latină, ceea ce și a fost prezentat în Capitolul 2. Obiectivele au constat în elaborarea unui algoritm eficient de transliterare, algoritmizarea procesului dat, implementarea algoritmului într-un sistem software și minimizarea erorii procesului de conversie de la grafia chirilică la cea latină.

În lucrare se abordează probleme ce apar în procesul de digitizare și recunoaștere lexicografică a textelor vechi românești, se argumentează necesitatea creării resurselor electronice specifice care caracterizează evoluția limbii române moderne. Se prezintă rezultatele statistice obținute la recunoașterea unui text românesc din secolul al XIX-lea, utilizându-se produse program moderne. Se propune o tehnologie în vederea creării lexiconului lingvistic pentru patrimoniul moldovenesc tipărit cu alfabet chirilic în perioada 1967-1989, pornind de la lexiconul românesc modern. Această tehnologie se bazează pe transliterare și pe aliniere paralelă a textelor.

Resursele digitizate sunt înregistrări specifice stocate într-o bază de date postată pe internet. Tehnologia propusă se axează pe soluționarea cu succes, pentru fiecare perioadă din evoluția limbii, a două probleme majore: **1**. Elaborarea (dezvoltarea) algoritmilor pentru recunoașterea literelor specifice perioadei; **2**. Elaborarea instrumentarului și interfețelor necesare pentru crearea resurselor lingvistice (lexiconului) corespunzător perioadei în scopul eficientizării procesului de recunoaștere a cuvintelor și de aliniere a normelor lingvistice contemporane.

La trecerea de la o perioadă la alta, în limitele posibilităților, se vor utiliza instrumentarul și resursele deja elaborate, materializând astfel principiul „din prezent în adâncul secolelor”. Resursele electronice create pot fi amplasate pe internet pentru acces public, contribuind la dezvoltarea mediului de comunicare informațională pentru limba română. În plus, aceste resurse ar constitui un suport esențial pentru cercetători, iar convertite în text literar ar putea fi utilizate ca materiale didactice în procesul de instruire.

Teza de licență abordează tematica implementării unui sistem unic de transliterare a mai multor tipuri de texte chirilice, ce sunt specifice mai multor perioade istorice. Așadar, realizarea unei aplicații software care va include mai mulți algoritmi de transliterare. De asemenea, realizarea unei interfețe grafice intuitive și ușor de utilizat. Baza teoretică și metodologică a tezei o constituie studierea lucrărilor fundamentale cu privire la specificul evoluției alfabetelor (slavon, chirilic și latin), analiza lucrărilor deja existente la tema dată. Implementarea algoritmilor realizați într-o unitate software pentru automatizarea procesului dat.

**Bibliografie**

1. D. Cristea, A. Iftene, A. Moruz, M. Moruz, „Semi-automatic alignment of old Romanian words using lexicons, In: Proceedings of the 8-th International Conference Linguistic resources and tools for processing of the Romanian language”, Iaşi, Editura Universităţii „A.I. Cuza”, 2012, p. 119-125.
2. G. Ivănescu, 2. Istoria limbii romane, Iaşi, 1980. [G. Ivănescu, History of the Romanian language, Iaşi, 1980.
3. Ştefan Munteanu şi Vasile Ţâra, Istoria limbii române literare, Editura Didactică şi Pedagogică, Bucureşti, 1978.
4. Cartea Moldovei (sec XVII – înc. sec XX). Ediţii cu caractere chirilice (sec XVII – înc. sec XX),Catalog general, Chişinău, 1992.
5. Zamﬁra Mihail, 155 cărţi într-o carte, Editura Prometeu, Chişinău, 2010, 532 p.
6. Valori Biblioﬁ le-2008, Rev. Gazeta bibliotecarului, Iunie-Iulie 2008, nr. 6-7, p.1 <http://87.248.191.115/bnrm/publicatii/ﬁles/3/93.pdf>
7. Gheţie I., Istoria limbii române literare, Bucureşti, 1978.
8. Cojocaru S., Ciubotaru C., Burlaca O., Colesnicov A., Verlan T., Malahov L., Petic M., Magariu G., Applications based on reusable linguistic resources. In Multilinguality and interoperability in language processing with emphasis on Romanian, Bucureşti, 2010, p.461-476.
9. Densuşianu, A., Istoria limbii şi literaturii române, Iaşi, 1894, <http://ru.scribd.com/doc/123035210/Istoria-limbii-si-literaturii-romane>
10. <https://ro.wikipedia.org/wiki/Transliterare>
11. Demidova V. „PARTICULARITĂŢILE DECHIRILIZĂRII LIMBII ROMÂNE” <http://oaji.net/articles/2015/2054-1446803236.pdf>