УДК 004.934.5

Ю.С. Гецэвіч, Б.М. Лабанаў, Д.А. Пакладок

ФАНЕТЫЧНАЯ І АЛАФОННАЯ АПРАЦОЎКА ТЭКСТУ Ў СІНТЭЗАТАРЫ БЕЛАРУСКАГА І РУСКАГА МАЎЛЕННЯ ДЛЯ МАБІЛЬНЫХ ПЛАТФОРМАЎ

Апісваюцца метады пераўтварэння «графема – фанема» і «фанема – алафон» для беларускай і русскай моў на аснове экспертных правіл. Прапануюцца алгарытмы пераўтварэння «графема – фанема» і «фанема – алафон» для сінтэзу маўлення па тэксце для мабільных платформаў.

Уводзіны

Фанетычны працэсар у складзе сістэмы сінтэзу маўлення па тэксце ажыццяўляе пераўтварэнне арфаграфічнага тэксту ў паслядоўнасць алафонаў. У фанетычным працэсары закладзены правілы пераўтварэння арфаграфічнага тэксту ў паслядоўнасць фанем (пераўтварэнне «графема – фанема») і правілы пераўтварэння паслядоўнасці фанем у алафонную паслядоўнасць (пераўтварэнне «фанема – алафон») [1].

На дадзены момант вылучаюць тры метады пераўтварэння «графема – фанема» [2]: пошук па слоўніку, метад на аснове экспертных правіл, метад кіруемых дадзеных.

Выкарыстанне слоўніка для запісу ўсіх фанетычных транскрыпцый словаформаў мовы дае найменшую вылічальную складанасць, аднак такі слоўнік займае шмат камп'ютарнай памяці. Асабліва гэта крытычна для флектыўных моў (да якіх належаць беларуская і руская мовы), таму што колькасць словаформаў у мове перавышае 2 млн.

Метад кіруемых дадзеных заснаваны на правілах, аднак у дадзеным выпадку гэтыя правілы выводзяцца алгарытмам на этапе навучання і аптымізуюцца для наступнай апрацоўкі з дапамогай выраўноўвання, дрэў, графаў, нейронавых сетак, статыстычных метадаў, часткова экспертам. Такім чынам здымаецца нагрузка на эксперта па вывадзе правіл і з'яўляецца магчымасць саманавучання сістэмы ў выніку папаўнення навучальнага корпуса. Дадзены метад з'яўляецца самым дакладным пры наяўнасці, па-першае, даволі вялікага корпусу фанетычна размечаных тэкстаў і дазволу на выкарыстанне амаль неабмежаваных вылічальных рэсурсаў і памяці для прымянення ў сінтэзе маўлення атрыманай колькасці правіл. Аднак пабудова вялікага корпусу, які быў бы размечаны фанемамі і алафонамі, з'яўляецца найбольшай складанасцю для выкарыстання метаду кіруемых правіл у сінтэзе маўлення.

Пераўтварэнне па правілах з'яўляецца класічнай алгарытмізацыяй ведаў пра мову. Выкарыстанне замацаваных правіл вымаўлення дае добрыя вынікі пры досыць невялікай колькасці правіл. Аднак у гэтым метадзе павялічваецца вылічальная складанасць адносна першага метаду, бо ў горшым выпадку для ўваходнага слова патрэбна прымяніць кожнае правіла пераўтварэння. Таксама ўскладаецца шмат абавязкаў на экспертаў-інжынераў па распрацоўцы зразумелых, аддзеленых ад алгарытмаў, проста рэдагуемых формаў запісу правіл, якія б маглі выкарыстоўваць эксперты-лінгвісты.

Практычнае выкарыстанне экспертных правіл пераўтварэння «графема – фанема» апісана для шматлікіх моў: літоўскай [3], польскай [4], грэцкай [5] і інш. Правілы пераўтварэння «графема – фанема – алафон» для рускага маўлення на цяперашні час фармалізаваныя [6] і рэалізаваныя без аддзялення ад праграмнага коду ў сінтэзатары маўлення Multiphone [1]. Таму яны не зручныя для хуткага рэдагавання і маштабавання экспертам-лінгвістам, калі знаходзяцца памылкі ў алафанізацыі слоў. Таксама гэтыя правілы з-за прывязкі да мовы праграмавання C++ не могуць быць адразу ж выкарыстаны для Java- і PHP-платформаў. Так як правілы пераўтварэння «графема – фанема – алафон» беларускага сінтэзатара маўлення былі перапрацаваны з правіл сінтэзатара рускага маўлення [7], то складанасць іх дапрацоўкі для беларускай мовы таксама прысутнічае.

На мабільныя камп'ютары накладаюцца абмежаванні на памер выкарыстання аператыўнай і пастаяннай памяці, а таксама на дазволеную колькасць працэсарных аперацый у адзінку часу [8]. Таму ў дадзенай працы прапануецца метад пераўтварэння «графема – фанема – алафон», які заснаваны на экспертных правілах. Таксама адзначаецца магчымасць выкарыстання гэтага падыходу і для стварэння фанетычнага працэсара сінтэзатара маўлення для інтэрнэт-платформаў. Для фармалізацыі форм правіл і саміх правіл фанетызацыі і алафанізацыі беларускіх слоў выкарыстоўвалася адпаведная літаратура [9–13].

1. Метад пераўтварэння «графема – фанема» для дзвюх моў

Найбольш распаўсюджаны падыход у рамках гэтага метаду заключаецца ў апрацоўцы сімвальнай паслядоўнасці злева направа, і для кожнага сімвала ўваходнай паслядоўнасці графем ужываецца адно або некалькі правіл для генерацыі фанемы. Відавочна, што гэтыя правілы не могуць працаваць ізалявана. У адваротным выпадку, напрыклад для рускай мовы, сімвал /*и*/ будзе заўсёды прыведзены да фанемы /I/. Таму замест аднаго сімвала /*и*/ аналізуецца яшчэ і як мінімум левы суседні сімвал, каб можна было згенераваць фанему /Y/, калі перад /*и*/ стаяць /*ж*/ або /*ш*/. У агульным выпадку гэтыя правілы маюць выгляд

$$A \rightarrow B / D / C$$

і азначаюць, што сімвал D будзе пераўтвораны ў A, калі левыя сімвалы – гэта B, а правыя – C.

У сістэме сінтэзу маўлення такія запісы адлюстроўваюць замацаваныя правілы вымаўлення ў мове. Заўважым, што парадак прымянення правіл з'яўляецца важным. Напрыклад, /*ть*/ павінна быць пераўтворана ў /T'/, у той час як /*ться*/ ў /C,C,A/. Адпаведна другое правіла павінна выконвацца раней за першае. Такое парадкаванне правіл ускладаецца на эксперта-лінгвіста.

Беларуская і руская мовы належаць да ўсходнеславянскай групы моў. Пісьмо і чытанне вядзецца злева направа. Для іх, як і для ўсіх славянскіх моў, характэрна наяўнасць двух эфектаў асіміляцыі папярэдняй зычнай фанемы: па глухасці-звонкасці, па цвёрдасці-мяккасці. Прычым неабходна заўважыць, што эфект асіміляцыі па глухасці-звонкасці можа быць унутрыслоўным і міжслоўным. Пры гэтым іх распаўсюджванне на суседнія графемы ідзе з процілеглага чытанню боку, гэта значыць справа налева. Так як пазначаныя эфекты не ўплываюць адзін на аднаго, то можна размежаваць метад пераўтварэння «графема – фанема» на чатыры паслядоўных этапы:

1) праверка графемы на адпаведнасць правілам, якія ўлічваюць кананічныя змены, а таксама праверка на эфекты асіміляцыі зычных фанем па глухасці-звонкасці, і замена ў выпадку супадзення на адпаведную фанему або групу фанем;

2) замена літары на фанему па «стандартных» правілах;

3) праверка мяккасці папярэдняй графемы (неабходная, але недастатковая ўмова для мяккасці);

4) праверка графемы на адпаведнасць правілам змякчэння і дадаванне мяккасці да дадзенай фанемы ў выпадку супадзення.

Такім чынам, прыведзены метад можна прымяніць для апрацоўкі дзвюх моў з дапамогай адпаведных пэўнай мове экспертных правіл. Інакш кажучы, ён з'яўляецца моваНезалежным, бо для апрацоўкі пэўнай мовы неабходна змяніць толькі ўваходныя рэсурсы (тэкст і фармалізаваныя правілы пераўтварэння «графема – фанема»).

Структура экспертных правіл складаецца з чатырох блокаў:

1. «Стандартныя» правілы замены графемы на фанему, г. зн. найбольш частотныя замены. Напрыклад, графема «А» у беларускай мове зачастую замяняецца на фанему «А».

2. Выключэнні з «стандартных» правіл замены ў выглядзе рэгулярных выразаў. Тут размяшчаюцца правілы, якія супярэчаць звычайнай замене. Гэта можа быць эфект асіміляцыі, канічная замена і г. д. Напрыклад, у рускай мове графема «И» пераходзіць у фанему «Ы», калі перад ёй стаіць графема «Ж», «Ш» ці «Ц».

3. Змякчальныя графемы, г. зн. тыя графемы, перад якімі можа з'яўляцца эфект змякчэння.

4. Правілы змякчэння у выглядзе рэгулярных выразаў. Тут апісваецца неабходная умова, пры якой дадзеная графема перайдзе у мяккую фанему.

Прыклады фармата фармалізаваных правіл пераўтварэння «графема – фанема» прадстаўлены ў табл. 1.

Табліца 1

Фрагменты правіл пераўтварэння «графема – фанема» для рускай і беларускай моў

Руская мова	Беларуская мова
#Общие правила буква-фонема	#Агульныя правілы графема-фанема
А-А	А-А
Б-В	Б-В
В-V	В-V
…	…
Ь-0	Э-Е
Э-Е	Ю-U
Ю-U	Я-А
Я-А	'-0
#Исключения из общих правил	#Выключэнні з агульных правілаў
(О)[+=]-О	[ГКХ][_\/](І)[_\/]-І
[ЖШЦ](И)-Y	[АЕЁІОУЫЭЮЯ][_\/](І)[_\/]-J'
…	…
(Л)[Н][Ц]-0	[Г](Д)[Т]-0
#Смягчающие графемы	#Змякчальныя графемы
Е	Е
…	…
Ь	Ь
#Общие правила смягчения	#Агульныя правілы змякчэння
([ХМНЛРБПВФДТЗСГК])[ЕЁЮЯИЬ]	([ХМНЛБПВФЗСГКЦ])[ЕЁЮЯІЬ]
…	…
([ГК])[ГКХ]	(С)[П]

2. Алгарытм пераўтварэння «графема – фанема»

Алгарытм пераўтварэння «графема – фанема» замяняе ўсе літары сінтагмы на іх фанетычныя аналагі і складаецца з сямі этапаў:

1) падрыхтоўкі дадзеных (*крокі 1–4*);
2) апрацоўкі службовых сімвалаў (*крокі 5, 8*);
3) фанемнага пераўтварэння па выключэнням з правіл (*крок 6*);
4) фанемнага пераўтварэння па «стандартным» правілам (*крок 7*);
5) праверкі мяккасці папярэдняй графемы (*крок 9*);
6) апрацоўкі змякчэння фанемы па правілам змякчэння (*крок 10*);
7) апрацоўкі вынікаў алгарытма (*крокі 11, 12*).

Уваходныя дадзеныя алгарытму:

Тэкст з прастаўленымі пазіцыямі націскаў і інтанацыйнымі меткамі *S*.

Рэсурсы алгарытму:

1. Мноства галосных фанем $Vow=\{Vow_1,\ldots,Vow_{NVow}\}$, дзе $NVow$ – колькасць галосных фанем; мноства абазначэнняў раздзяляльнікаў $D=\{D_1,\ldots,D_{ND}\}$ (заўважым, што D_1 з'яўляецца сімвалам пачатку або канца сінтагмы), дзе ND – колькасць абазначэнняў раздзяляльнікаў; мноства абазначэнняў націскаў $Sstr=\{Sstr_1,\ldots,Sstr_{NSstr}\}$, дзе $NSstr$ – колькасць абазначэнняў націскаў; службовы сімвал змякчэння $Soft$=/'/.

2. Правілы «графема – фанема» $HV=\{<H_1,V_1>,\ldots,<H_{NHV},V_{NHV}>\}$, дзе NHV – колькасць сімвалаў мовы $H=\{H_1,\ldots,H_{NHV}\}$ і адпаведных фанем у мове $V=\{V_1,\ldots,V_{NHV}\}$; выключэнні з правіл «графема – фанема» $RF=\{<R_1,F_1>,\ldots,<R_{NRF},F_{NRF}>\}$, дзе NRF – колькасць пар правіл

выключэнняў у выглядзе рэгулярных выразаў $R=\{R_1,\ldots,R_{NRF}\}$ і адпаведныя ім фанемы $F=\{F_1,\ldots,F_{NRF}\}$; спіс змякчальных фанем $M=\{M_1,\ldots,M_{NM}\}$, дзе NM – колькасць змякчальных фанем; правілы змякчэння ў выглядзе рэгулярных выразаў $E=\{E_1,\ldots,E_{NE}\}$, дзе NE – колькасць правіл змякчэння суседняй фанемы.

Выхадныя дадзеныя:

1. Паслядоўнасць фанетычных сінтагмаў $Sph=\bigcup_{i=1}^{NSph} Sph_i$, $Sph_i=\bigcup_{j=1}^{NPh} Ph_j$, $i = 1, \ldots, NSph$.

2. Паслядоўнасць колькасці складоў у кожнай фанетычнай сінтагме $NSyl=\bigcup_{i=1}^{NSph} Nsyl_i$.

Спецыяльныя функцыі:

Regex(t, r) – функцыя, якая ажыццяўляе пошук рэгулярнага выразу r у тэкставым радку t.

$$\text{Regex}(t, r)=\begin{cases}0, \text{калі } t \text{ не адпавядае рэгулярнаму выразу } r;\\ 1, \text{калі } t \text{ не адпавядае рэгулярнаму выразу } r.\end{cases}$$

Алгарытм:

Уваход:

Крок 1. Лічым, што $S=\bigcup_{i=1}^{NS} S_i$, дзе NS – колькасць сінтагмаў уваходнага тэксту.

Крок 2. Для кожнай сінтагмы S_i, $i = 1, \ldots, NS$ выконваем крокі 3–12. Далей *крок 13*.

Крок 3. Лічым індыкатар мяккасці роўным нулю, гэта значыць $Flag$:=0.

Крок 4. Для кожнай паслядоўнасці сімвалаў X_{j-2}, X_{j-1}, X_j, X_{j+1}, X_{j+2}, $j = NX, \ldots, 1$ з сінтагмы S_i ($S_i=\bigcup_{j=1}^{NX} X_j$, NX – колькасць сімвалаў у сінтагме) выконваем *крокі 5–11*. Прычым павінна выконвацца сістэма ўмоў:

$$\begin{cases} X_{j-2} := NULL, j-2<1;\\ X_{j-1} := NULL, j-1<1;\\ X_{j+1} := NULL, j+1>NX;\\ X_{j+2} := NULL, j+2>NX.\end{cases}$$

Калі сімвалы ў дадзенай сінтагме S_i скончыліся, пераходзім да *кроку 2*.

Крок 5. Калі сімвал X_j з'яўляецца раздзяляльным ці абазначэннем націску, г. зн. $X_{j+1}\in(D\cup Sstr)$, то лічым $TEMP$:=X_j і пераходзім да *кроку 11*.

Крок 6. Для кожнага рэгулярнага выразу R_k, $k = \{1, ..., NRF\}$ вылічыць $\text{Regex}(X_{j-2}\cup X_{j-1}\cup X_j\cup X_{j+1}\cup X_{j+2}, R_k)$. Калі для некаторага k умова $\text{Regex}(X_{j-2}\cup X_{j-1}\cup X_j\cup X_{j+1}\cup X_{j+2}, R_k)\neq 0$ выконваецца, то лічым $TEMP$:=F_k, і далей *крок 8*.

Крок 7. Для кожнага H_m, $m = 1, \ldots, NHV$ правяраем, ці супаў сімвал мовы H_m з наяўным сімвалам X_j. Калі для некаторага m умова $H_m = X_j$ была выканана, то знаходзім адпаведную ёй фанему з пары $<H_m,V_m>$ і лічым $TEMP$:=V_m.

Крок 8. Калі бягучая фанема будзе роўная пустому мноству $TEMP$=NULL, то перайсці да *кроку 4*.

Крок 9. Калі сімвал X_{j+1} змякчальны, г. зн. $X_{j+1}\in M$ або $Flag$:=1, то перайсці да *кроку 10*. Інакш лічым $Flag$ = 0 і пераходзім да *кроку 11*.

Крок 10. Для кожнага правіла змякчэння E_n, $n = 1, ..., NE$, вылічваем $\text{Regex}(X_j\cup X_{j+1}, E_n)$. Калі для некаторага n умова $\text{Regex}(X_j\cup X_{j+1}, E_n)\neq 0$ выконваецца,

то дадаем сімвал змякчэння да фанемы *TEMP*, г. зн. лічым $TEMP := TEMP \cup Soft$, таксама мяняем індыкатар мяккасці *Flag*:=1.

Крок 11. Дадаванне *TEMP* да выніковай паслядоўнасці фанем сінтагмы Sph_i знойдзенай фанемы *TEMP*, а менавіта $Sph_i := TEMP \cup Sph_i$. Калі $TEMP \in Vow$, то $Nsyl_i$:= $Nsyl_i$ + 1, дзе $Nsyl_i$ – колькасць складоў у сінтагме Sph_i.

Крок 12. Дадаванне фанетычнай сінтагмы Sph_i да выхаднога фанетычнага тэксту *Sph*, г. зн. $Sph := Sph \cup Sph_i$. Захаванне колькасці складоў у сінтагме Sph_i у паслядоўнасці *Nsyl*, г. зн. $Nsyl := Nsyl \cup Nsyl_i$, што абазначае колькасць складоў адпаведна кожнай раней апрацаванай сінтагме.

Крок 13. Канец алгарытму.

3. Метад пераўтварэння «фанема – алафон» для дзвюх моў

Дадзенае пераўтварэнне ажыццяўляе замену фанемы на кадыфікаваную пазнаку алафона. Алафон – натуральны гук, які абумоўлены пэўным фанетычным асяроддзем. У адрозненне ад фанемы алафон з'яўляецца не абстрактным паняццем, а пэўным маўленчым гукам. У акустычным працэсары кадыфікаваная пазнака алафона будзе ўказваць на адпаведны алафон з акустычнай базы, які будзе мадыфікавацца і склейвацца з іншымі алафонамі ў адзін алафонны радок – маўленчы сігнал. Для спрошчанага апісання працэсаў утварэння гуку ў артыкуле будзем ўсюды выкарыстоўваць паняцце алафон замест выразу *кадыфікаваная пазнака алафона*.

У якасці рэалізацыі агульнага алгарытму пераўтварэння «фанема – алафон» для мабільных прыстасаванняў была задзейнічана спрошчаная экспертная сістэма: правілы запісаны ў выглядзе класіфікацыі груп цэнтральнага элемента, левага і правага кантэксту, а таксама выхаднога значэння цэнтральных груп фанем для ўсіх магчымых варыяцый кантэкстаў. Для дасягнення аптымальных суадносін прадукцыйнасці і спажывання рэсурсаў экспертныя правілы пераўтвараюцца сістэмай у машынныя хэш-коды падчас ініцыялізацыі сінтэзу маўлення. Такая схема дазваляе пазбегнуць разбору і захоўвання значэнняў для кожнага элемента пры ўсіх магчымых варыяцыях кантэкстаў пры невялікім павелічэнні вылічальнай складанасці алгарытму, г. зн. памяншэнне выкарыстання аператыўнай памяці на 70 % пры павелічэнні на 5–10 дадатковых ітэрацый алгарытма пошуку адпаведнага кантэксту для алафона.

Правілы пераўтварэння «фанема – алафон» уяўляюць сабой вынік эксперыментальнага даследавання розных фанетычных груп па іх фанетычным асяроддзі. Пасля аналізу тэарэтычных выкладак [1, 3] і практычнай рэалізацыі літарафанемаалафоннага працэсара сістэмы сінтэзу маўлення Multiphone быў распрацаваны зручны фармат запісу правіл пераўтварэнняў «фанема – алафон». Характэрна, што такі фармат правіл дазваляе адначасова запісваць і выкарыстоўваць правілы пераўтварэння як для беларускай, так і для рускай мовы (табл. 2).

Структура экспертных правіл складаецца з наступных блокаў:

1) алфавіту сімвалаў, якія могуць утвараць алафоны. Уключае ў сябе фанемы мовы і сімвал пачатку ці канца сінтагмы;

2) цэнтральных груп фанем (фанетычных груп, характэрных для пэўнай цэнтральнай фанемы);

3) груп фанем левага кантэксту, якія могуць уплываць на фанетычнае асяроддзе пэўнай цэнтральнай фанемы злева;

4) груп фанем правага кантэксту, якія могуць уплываць на фанетычнае асяроддзе пэўнай цэнтральнай фанемы зправа;

5) правілаў пераўтварэння для кожнай цэнтральнай групы фанем, якія суадносяць для цэнтральнай групы фанем пэўныя значэнні левага і правага кантэкстаў алафонаў.

Табліца 2

Правілы пераўтварэння «фанема – алафон» для дзвюх моў (фрагменты)

Назва групы	Значэнні (скарочана)
Алфавіт сімвалаў	A0,A1,A2,A3,E0,E1,E2,E3,I0,I1,I2,I3,O0,O1,O2,O3,Y0,Y1,Y2,Y3,U0, U1,U2,U3,B,B',D,D',G,G',Z,Z',ZH,L,L',M,M',N,N',P,P',T,T',K,K',C,CH', F,F',S,S',H,H',SH,SH',V,V',J',R,R',C',CH,SCH,GH,DZ',DZH,GH',W,#,
Цэнтральныя групы фанем	C0:,A0,A1,A2,A3,E0,E1,E2,E3,I0,I1,I2,I3,Y0,Y1,Y2,Y3,
	C1:,U0,U1,U2,U3,O0,O1,O2,O3,
	…
	C6:,R,R',
Групы фанем левага кантэксту	L0:,#,
	L1:,P,B,F,V,M,U0,U1,U2,U3,O0,O1,O2,O3,W,
	L2:,SH,ZH,R,T,C,S,D,Z,N,L,A0,A1,A2,A3,E0,E1,E2,E3,Y0,Y1,Y2,Y3, CH,SCH,DZH,
	…
	L11:,B,B',D,D',G,G',Z,Z',ZH,L,L',M,M',N,N',P,P',T,T',K,K',C,CH',F,F',S, S',SH,SH',H,H',V,V',J',R,R',C',CH,SCH,GH,DZ',DZH,GH',W,#,
Групы фанем правага кантэксту	R0:,#,
	R1:,P,B,F,V,M,L,W,O0,O1,O2,O3,U0,U1,U2,U3,
	…
	R14:,A0,A1,A2,A3,E0,E1,E2,E3,Y0,Y1,Y2,Y3,I0,I1,I2,I3,O0,O1,O2, O3,U0,U1,U2,U3,
Правілы пераўтварэння для кожнай цэнтральнай групы фанем	C0
	L0R0-00
	L5R0-10
	L6R0-20
	…
	C1
	L0R0-00
	L1R0-10
	L2R0-20
	…

4. Алгарытм пераўтварэння «фанема – алафон»

Алгарытм пераўтварэння «фанема – алафон» складаецца з сямі этапаў:

1) падрыхтоўкі дадзеных (*крокі 1–4*);
2) вызначэння першага індэкса алафона *(крокі 5–11*);
3) апрацоўкі службовых сімвалаў (*крок 12*);
4) вызначэння мноства фанемных груп для цэнтральнай, правай і левай фанем (*крок 13*);
5) пошука такога варыянта сярод правіл для вызначанай фанемнай групы цэнтральнай фанемы, пры якім існуе пэўнае значэнне левага і правага кантэкстаў алафона – другі і трэці індэксы алафона (*крокі 14–16*);
6) знаходжання поўнага значэння алафона (*крок 17*);
7) апрацоўкі вынікаў алгарытма (*крокі 18, 19*).

Уваходныя дадзеныя алгарытма:

1. Паслядоўнасць фанетычных транскрыпцый сінтагмаў уваходнага тэксту *Sph*.
2. Паслядоўнасць колькасцяў складоў у кожнай сінтагме *Nsyl*.

Рэсурсы алгарытма:

1. Мноства галосных фанем $Vow=\{Vow_1,\ldots,Vow_{NVow}\}$, дзе *NVow* – колькасць галосных фанем; мноства зычных фанем $Con=\{Con_1,\ldots,Con_{NCon}\}$, дзе *NCon* – колькасць зычных фанем; мноства раздзяляльнікаў $D=\{D_1,\ldots,D_{ND}\}$ (заўважым, што D_1 з'яўляецца знакам пачатку або

канца сінтагмы), дзе *ND* – колькасць раздзяляльнікаў; мноства абазначэнняў націскаў $Sstr = \{Sstr_1, \ldots, Sstr_{NSstr}\}$, дзе *NSstr* – колькасць абазначэнняў націскаў, $Sstr_1$ – сімвал поўнага націску, $Sstr_2$ – сімвал частковага націску.

2. Фанемныя групы для цэнтральных фанем $C = \{<NameC_1, C_1>, \ldots, <NameC_{NC}, C_{NC}>\}$, $m = 1, \ldots, NC$, дзе $C_m = \{PhC_{m,r} \in \{Vow \cup Con\} \mid r = 1,\ldots, NC_m\}$, C_m – мноства фанем групы, $NameC_m$ – унікальная назва групы. Заўважым, што фанемы $PhC_{m,r}$ выкарыстоўваюцца ў *C* не больш аднаго разу.

Фанемныя групы для левых фанем $L = \{<NameL_1, L_1>, \ldots, <NameL_1, L_{NL}>\}$, $n = 1, \ldots, NL$, дзе $L_n = \{PhL_{n,s} \in \{Vow \cup Con \cup D\} \mid s = 1,\ldots, NL_n\}$, L_n – мноства фанем групы, $NameL_n$ – унікальная назва групы.

Фанемныя групы для правых фанем $R=\{<NameR_1,R_1>,\ldots,<NameR_{NR},R_{NR}>\}$, $p=1,\ldots,NR$, дзе $R_p = \{PhR_{p,t} \in \{Vow \cup Con \cup D\} \mid t = 1,\ldots, NR_p\}$, R_p – мноства фанем групы, $NameR_p$ – унікальная назва групы.

Заўважым, што кожная група фанем мае ўнікальную назву, напрыклад «C2», «L1», «R13». Першы элемент назвы адлюстроўвае прыналежнасць да месца размяшчэння фанемы (C – групы для цэнтральных фанем, L – групы для левых фанем, R – групы для правых фанем), другі вызначае склад фанем, якія ўваходзяць у дадзеную групу.

3. Для кожнай фанемнай групы C_m, $m = 1, \ldots, NC$, вызначаны правілы пераўтварэння «фанема – алафон» $RA_m = \{<LR_1, Index_1>, \ldots, <LR_{NRAm}, Index_{NRAm}>\}$, *NRAm* – колькасць пар элементаў правіл $LR = \{LR_1, \ldots, LR_{NRAm}\}$ і $Index = \{Index_1, \ldots, Index_{NRAm}\}$.

Выхадныя дадзеныя:

1. Паслядоўнасць алафонных транскрыпцый сінтагмаў *Sal*.

Функцыя $F(LR_i, NameL_j, NameR_k)$ правярае, ці роўны першы элемент правіл пераўтварэння «фанема – алафон» LR_i аб'яднанню назваў груп для левых $NameL_j$ і правых $NameR_k$ фанем:

$$F(LR_i, NameL_j, NameR_k) = \begin{cases} 0, LR_i \neq NameL_j \cup NameR_k; \\ 1, LR_i = NameL_j \cup NameR_k. \end{cases}$$

Алгарытм:

Крок 1. Лічым, што $Sph = \bigcup_{i=1}^{NSph} Sph_i$, $Nsyl = \bigcup_{i=1}^{NSph} Nsyl_i$, *NSph* – колькасць сінтагмаў з распісанымі фанетычнымі транскрыпцыямі.

Крок 2. Для кожнай сінтагмы ў фанемным выглядзе Sph_i, $i = 1, \ldots, NSph$ (заўважым, што $Sph_i = \bigcup_{j=1}^{NPh} Ph_j$, *NPh* – колькасць фанем у сінтагме) выконваем *крокі 3–19*. Далей *крок 20*.

Крок 3. Лічым наяўную колькасць складоў у сінтагме Sph_i роўнай *NsylT*:=0;

Крок 4. Для кожнай паслядоўнасці фанем Ph_{j-1}, Ph_j, Ph_{j+1} сінтагмы Sph_i выконваем крокі 5–18, прычым $j = NPh, \ldots, 1$. Пры гэтым павінна выконвацца сістэма ўмоў:

$$\begin{cases} Ph_{j-1} := D_1, j - 1 < 1; \\ Ph_{j+1} := D_1, j + 1 > NPh. \end{cases}$$

Калі фанемы ў дадзенай сінтагме скончыліся, пераходзім да *кроку 2*.

Крок 5. Калі фанема $Ph_j \in Vow$, то пераходзім да *кроку 6*. Інакш пераходзім да *кроку 10*.

Крок 6. Павялічваем колькасць складоў у сінтагме Sph_i на адзінку *NsylT*:=*NsylTI*+1.

Крок 7. Калі $Ph_{j+1} = Sstr_1$ ($Sstr_1$ = /+/), то дадзеная фанема мае поўны націск, тады першы індэкс алафона *IndexA* прымае значэнне 0, а менавіта *IndexA*:=0. Тады за фанему Ph_{j+1} прымаем фанему з індэксам *j*+2 з паслядоўнасці фанем Sph_i, а ў паслядоўнасці фанем Sph_i выдаляем фанему з індэксам *j*+1, агульную колькасць фанем памяншаем на адзінку ў паслядоўнасці фанем Sph_i *NPh*:=*NPh*–1. Запамінаем, што наяўная галосная Ph_j – націскная праз *FlagSV*:= 1, і пераходзім да *кроку 13*.

Крок 8. Калі Ph_{j+1} = $Sstr_2$ ($Sstr_2$ = /=/), то дадзеная фанема мае частковы націск, тады першы індэкс алафона *IndexA* прымае значэнне 1, а менавіта *IndexA*:=1. Тады за фанему Ph_{j+1} прымаем фанему з індэксам *j*+2 з паслядоўнасці фанем Sph_i, а ў паслядоўнасці фанем Sph_i выдаляем фанему з індэксам *j*+1, агульную колькасць фанем памяншаем на адзінку ў паслядоўнасці фанем Sph_i *NPh*:=*NPh*–1. Запамінаем, што наяўная галосная Ph_j – націскная праз *FlagSV*:= 1, і пераходзім да *кроку 13*.

Крок 9. Калі *FlagSV* = 0 і дадзеная фанема Ph_j не знаходзіцца ў крайнім левым $NsylT \neq 1$ ці крайнім правым $NsylT \neq Nsyl_i$ складзе, то дадзеная фанема Ph_j з'яўляецца ненаціскной галоснай *другой* ступені рэдукцыі, тады першы індэкс алафона *IndexA* абазначаем лічбай 3, а менавіта *IndexA*:=3. Калі інакш, яна з'яўляецца ненаціскной галоснай *першай* ступені рэдукцыі, тады першы індэкс алафона *IndexA* абазначаем лічбай 2, а менавіта *IndexA*:=2. Пераходзім да *кроку 13*.

Крок 10. Калі дадзеная фанема зычная, г. зн. $Ph_j \in Con$, то пераходзім да *кроку 11*. Інакш пераходзім да *кроку 12*.

Крок 11. Калі фанема Ph_j роўная першай левай фанеме Ph_{j-1} адносна яе (Ph_j = Ph_{j-1}), то яна лічыцца падвоенай, г. зн. *IndexA*:=1. Тады за фанему Ph_{j-1} прымаем фанему з індэксам *j*–2 з паслядоўнасці фанем Sph_i, а ў паслядоўнасці фанем Sph_i выдаляем фанему з індэксам *j*–1, агульную нумарацыю наяўнай фанемы памяншаем на 1, г.зн. *j*:=*j*–1, бо агульная колькасць фанем паменшылася на адзінку ў паслядоўнасці фанем Sph_i *NPh*:=*NPh*–1.

Калі інакш, дадзеную фанему лічым адзінарнай *IndexA*:=0.

Пераходзім да *кроку 13*.

Крок 12. Калі Ph_j з'яўляецца раздзяляльным элементам $Ph_j \in D$, то запамінаем яго ў часовай зменнай *TEMP*:=Ph_j і пераходзім да *кроку 18*.

Крок 13. Вызначаем мноства фанемных груп, у якія ўваходзіць цэнтральны элемент Ph_j:

$$Cx = \{< NameC_m, C_m > | \; PhC_{m,r} \in C_m, C_m \in C, m = 1,..., NC, r = 1,..., NC_m, \text{калі } PhC_{m,r} = Ph_j \}.$$

Улічваючы фармат правіл пераўтварэння «графема – фанема», мноства *Cx* складаецца з аднаго элемента. Запамінаем індэкс *m*, пры якім была выканана ўмова $PhC_{m,r} = Ph_j$, г. зн. *iCx*:=*m*. Па індэксе *iCx* знаходзім адпаведныя гэтаму мноству правілы пераўтварэння RA_{iCx}.

Вызначаем мноства фанемных груп, у якія ўваходзіць левы элемент Ph_{j-1}:

$$Lx = \{< NameL_n, L_n > | \; PhL_{n,s} \in L_n, L_n \in L, n = 1,..., NL, s = 1,..., NL_n, \text{пры ўмове } PhL_{n,s} = Ph_j \}.$$

Вызначаем мноства фанемных груп, у якія ўваходзіць правы элемент Ph_{j+1}:

$$Rx = \{< NameR_p, R_p > | \; PhR_{p,t} \in R_p, R_p \in R, p = 1,..., NR, t = 1,..., NR_p, \text{пры ўмове } PhR_{p,t} = Ph_j \}.$$

Крок 14. Для кожнага мноства фанем знойдзенага левага кантэксту Lx_{k1}, $Lx = \{Lx_1, \ldots, Lx_{NLx}\}$, *k1* = 1, …, *NLx*, дзе *NLx* – колькасць груп для левага кантэксту Ph_{j-1}, выконваем *крокі 15–16*.

Крок 15. Для кожнага мноства фанем правага кантэксту Rx_{k2}, $Rx = \{Rx_1, \ldots, R_{NRx}\}$, *k2* = 1, …, *NRx*, дзе *NRx* – колькасць груп для правага кантэксту Ph_{j+1}, выконваем *крок 16*.

Крок 16. Для першага элемента LR_q кожнага правіла RA_{iCx}, *q* = 1, ..., NRA_{iCx} правяраем значэнне функцыі F(LR_q,$NameL_{k1}$,$NameR_{k2}$). Калі F(LR_q,$NameL_{k1}$,$NameR_{k2}$) = 1, то індэкс *q* вызначае адпаведны другі элемент правіла RA_{iCx} – $Index_q$. Гэты элемент $Index_q$ з'яўляецца другім і трэцім індэксам і алафона *IndexB*:=$Index_q$.

Крок 17. Знаходзім поўнае значэнне алафона $TEMP := Ph_j \cup IndexA \cup IndexB$.

Крок 18. Дадаём *TEMP* да алафоннай транскрыпцыі сінтагмы $Sal_i := TEMP \cup Sal_i$ і пераходзім да *кроку 4*.

Крок 19. Дадаём алафонную транскрыпцыю сінтагмы да выхаднога алафоннага тэксту $Sal := Sal \cup Sal_i$ і пераходзім да *кроку 2*.

Крок 20. Канец алгарытму.

5. Вынікі працы алгарытмаў «графема – фанема» і «фанема – алафон» для беларускай і рускай моў

Прывядзём некаторыя прыклады пераўтварэння «графема – фанема» і «фанема – алафон» для беларускай і рускай моў (табл. 3, 4).

Табліца 3

Прыклады літарафанемаалафоннага пераўтварэння слоў для беларускай мовы

Словы	Пераўтварэнне «графема – фанема»	Пераўтварэнне «фанема – алафон»
бюльбю+левы	B',U,L',B',U,+,L',E,V,Y	B'004,U243,L'001,B'002,U043,L'004,E341,V012,Y210
вэ+ндзіць	V,E,+,N',DZ',I,C'	V001,E013,N'001,DZ'004,I243,C'000
удзвю+х	U,DZ',V',U,+,H	U203,DZ'001,V'001,U042,H000
льві+ца	L',V',I,+,C,A	L'001,V'001,I042,C002,A220
міжго+р'е	M',I,ZH,GH,O,+,R,J',E	M'004,I242,ZH001,GH001,O032,R001,J'002,E240
геадэ+зія	GH',E,A,D,E,+,Z',I,J',A	GH'004,E242,A222,D002,E023,Z'004,I343,J'012,A240
суддзя+	S,U,DZ',DZ',A,+	S002,U223,DZ'102,A040
джу+нглі	DZH,U,+,N,GH,L',I	DZH002,U022,N001,GH004,L'004,I240
еўразо+на	J',E,W,R,A,Z,O,+,N,A	J'002,E241,W013,R012,A222,Z002,O022,N004,A220
ззя+нне	Z',Z',A,+,N',E	Z'102,A043,N'104,E240
касне+шся	K,A,S',N',E,+,S',S',A	K004,A233,S'002,N'002,E043,S'102,A240
і=ншакраі+нец	J',I,=,N,SH,A,K,R,A,J',I,+,N',E,C	J'002,I142,N003,SH002,A322,K004,R022,A223,J'011,I043,N'004,E242,C000

Табліца 4

Прыклады літарафанемаалафоннага пераўтварэння слоў для рускай мовы

Словы	Пераўтварэнне «графема – фанема»	Пераўтварэнне «фанема – алафон»
мужичо+чек	M,U,ZH,Y,CH',O,+,CH',E,K	M004,U212,ZH004,Y223,CH'001,O043,CH'002,E242,K000
кири+ллица	K',I,R',I,+, L',L',I,C,A	K'002,I243,R'002,I043,L'104,I342,C002,A220
объе+зд	A,B,J',E,+,S,T	A201,B001,J'001,E042,S002,T000
бе+лого	B',E,+,L,A,V,A	B'002,E041,L004,A311,V012,A210
проезжа+ться	P,R,A,J',E,ZH,A,+,C,C,A	P002,R012,A223,J'012,E242,ZH102,A022,C102,A220
со+лнце	S,O,+,N,C,E	S001,O022,N003,C002,E220
расчи+тывать	R,A,SH',I,+,T,Y,V,A,T'	R022,A223,SH'001,I042,T002,Y321,V012,A213,T'000
безотчё+тен	B',E,Z,A,CH',CH',O,+,T',E,N	B'004,E242,Z004,A223,CH'101,O043,T'002,E242,N000
разбе+жка	R,A,Z,B',E,+,SH,K,A	R022,A222,Z001,B'002,E042,SH002,K004,A230
ию+льский	I,J',U,+,L',S,K',I,J'	I203,J'011,U043,L'003,S002,K'002,I243,J'010

Пасля таго як алгарытмы былі рэалізаваны праграмна ў мабільным сінтэзатары маўлення, яны былі пратэставаны на корпусе слоў (табл. 5). Корпус слоў быў складзены паводле выбаркі ўсіх магчымых пяці сімвальных камбінацый сімвалаў у словах электронных слоўнікаў [14]. Спачатку гэты корпус быў апрацаваны стацыянарнай версіяй сінтэзатара маўлення Multiphone, а пасля – мабільнай версіяй. За эталон правільнасці быў прыняты вынік працы стацыянарнага сінтэзатара маўлення. Адносна яго была падлічана колькасць адпаведна для беларускай і рускай моў правільна апрацаваных мабільнай версіяй сінтэзатараў слоў – 82010 і 83990, фанем – 918440 і 888950, алафонаў – 903216 і 871989.

Табліца 5

Вынікі тэставання алгарытмаў для беларускай і рускай моў

Параметры	Беларуская мова	Руская мова
Колькасць слоў	102 169	101 958
Колькасць правільных слоў	**82 010**	**83 990**
Працэнтныя адносіны карэктна пераўтвораных слоў	80,269 %	82,377 %
Колькасць фанем	934 583	897 635
Колькасць правільных фанем	**918 440**	**888 950**
Працэнтныя адносіны карэктна пераўтвораных фанем	98,273 %	99,032 %
Колькасць алафонаў	934 583	897 635
Колькасць правільных алафонаў	**903 216**	**871 989**
Працэнтныя адносіны карэктна пераўтвораных алафонаў	96,644 %	97,143 %

Вынікі тэставання хуткасці алгарытмаў для беларускай і рускай моў на адабраных эскпертам-лінгвістам карпусах слоў аб'ёмам каля 1000 слоў прадстаўлены ў табл. 6. У корпусе былі прадстаўлены найбольш часта ўжывальныя словы для кожнай з моў. Як бачна з табліцы, хуткасць алгарытма «графема – фанема» вельмі залежыць ад фанетычнага склада слова. Алгарытм «фанема – алафон», наадварот, не залежыць ад слова і працуе вельмі хутка.

Табліца 6

Вынікі тэставання хуткасці алгарытмаў для беларускай і рускай моў

Алгарытм	Характарыстыка	Беларуская мова	Руская мова
«Графема – фанема»	Найменшы час, мс	15	15
	Найбольшы час, мс	80	72
	Сярэдні час, мс	35	33
«Фанема – алафон»	Найменшы час, мс	1	1
	Найбольшы час, мс	2	2
	Сярэдні час, мс	1	1
«Графема – фанема – алафонная апрацоўка»	Найменшы час, мс	16	16
	Найбольшы час, мс	81	73
	Сярэдні час, мс	36	34

Заключэнне

У артыкуле былі прапанаваны метады пераўтварэнняў «графема – фанема» і «фанема – алафон», якія заснаваны на экспертных правілах, для сінтэзу беларускага і рускага маўленняў. Прыведзеныя алгарытмы праверaны адносна эталоннай стацыянарнай сістэмы сінтэзу маўлення Мультыфон. Вынікі тэстаў сведчаць пра высокі ўзровень дакладнасці працы алгарытмаў для беларускай і рускай мовы пры апрацоўцы слоў (больш 80 %), фанем (больш 98 %), алафонаў (больш 96 %). Далейшая праца будзе накіравана на дапрацоўку практычнай рэалізацыі прыведзеных алгарытмаў.

Спіс літаратуры

1. Лобанов, Б.М. Компьютерный синтез и клонирование речи / Б.М. Лобанов, Л.И. Цирульник. – Минск : Беларус. навука, 2008. – 344 с.

2. Taylor, P. Text-to-Speech Synthesis / P. Taylor. – N.Y. : Cambridge University Press, 2009. – 626 p.

3. Norkevičius, G. Knowledge-based grapheme-to-phoneme conversion of Lithuanian words / G. Norkevičius, G. Raškinis, A. Kazlauskienė // SPECOM 2005, 10th Intern. Conf. Speech and Computer. – 2005. – P. 235-238.

4. Steffen-Batóg, M. An algorithm for phonetic transcription of orthographic texts in Polish / M. Steffen-Batóg, P. Nowakowski // Studia Phonetica Posnaniensia / eds. M. Steffen-Batóg, W. Awedyk. – Poznań : Wydawnictwo Naukowe UAM, 1993. – Vol. 3. – P. 135–183.

5. Chalamandaris, A. Rule-based grapheme-to-phoneme method for the Greek / A. Chalamandaris, S. Raptis, P. Tsiakoulis // 9th European Conference on Speech Communication and Technology. – 2005. – P. 2937–2940.

6. Цирульник, Л.И. Алгоритм генерации фонемной последовательности по орфографическому тексту в системе синтеза речи / Л.И. Цирульник // Информатика. – 2006. – № 4. – С. 61–70.

7. Гецевич, Ю.С. Система синтеза белорусской речи по тексту / Ю.С. Гецевич, Б.М. Лобанов // Речевые технологии. – 2010. – № 1. – С. 91–100.

8. Гецэвіч, Ю.С. Кампаненты для розных платформаў сінтэзатара маўлення па тэксце для інтэлектуальных сістэм / Ю.С. Гецэвіч, Д.А. Пакладок, Д.В. Брэк // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS–2013) : материалы III Междунар. науч.-техн. конф. (Минск, 21–23 февр. 2013 г.) / редкол. : В.В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2013 г. – С. 375–382.

9. Кароткая граматыка беларускай мовы. У 2 ч. Ч. 1. Фаналогія. Марфаналогія. Марфалогія. – Мінск : Беларус. навука, 2007. – 351 с.

10. Беларуская граматыка. У 2 ч. Ч. 1. Фаналогія. Арфаграфія. Марфалогія. Словаўтварэнне. Націск / АН БССР, Інстытут мовазнаўства імя Я.Коласа. – Мінск : Навука і тэхніка, 1985. – С 117–133.

11. Сучасная беларуская мова. Фанетыка. Фаналогія. Арфаэпія. Графіка. Арфаграфія : вучэбна-метадычны комплекс для студэнтаў 1-га курса спецыяльнасцей Г.10.02.01; Г.02.01.00 «Беларуская мова і літаратура». – Мінск : БДУ, 2002. – 144 с.

12. Янкоўскі, Ф.М. Беларускае літаратурнае вымаўленне / Ф.М. Янкоўскі. – Мінск : Народная асвета, 1976. – 91 с.

13. Закон Рэспублікі Беларусь, 23 ліпеня 2008 г., № 420-З. Аб Правілах беларускай арфаграфіі і пунктуацыі [Электронны рэсурс]. – 2013. – Рэжым доступу : http://www.pravo.by/main.aspx?guid=3871&p0=H10800420&p2. – Дата доступу : 02.07.2013.

14. Hetsevich, Y. Overview of Belarusian And Russian dictionaries and their adaptation for NooJ / Y. Hetsevich, S. Hetsevich // Automatic Processing of Various Levels of Linguistic Phenomena: Selected Papers from the NooJ 2011 Intern. Conf. / eds. Vučković Kristina, Bekavac Božo, Silberztein Max. – Newcastle : Cambridge Scholars Publishing, 2012. – P. 29–40.

Паступіла 25.11.2013

Аб'яднаны інстытут праблем
інфарматыкі НАН Беларусі,
Мінск, Сурганава, 6
e-mail: yury.hetsevich@gmail.com,
lobanov@newman.bas-net.by,
dima.pokladok@gmail.com.

Yu.S. Hetsevich, B.M. Lobanov, D.A. Pokladok

PHONETIC AND ALLOPHONIC TEXT PROCESSING IN BELARUSIAN AND RUSSIAN SPEECH SYNTHESIZER FOR MOBILE PLATFORMS

The article describes methods of «grapheme – phoneme» and «phoneme – allophone» conversions for Belarusian and Russian speech synthesis. For speech synthesizers on mobile platforms, the rule-based method has been selected. The article describes text processing algorithms at the input phase and rules for «grapheme – phoneme» and «phoneme – allophone» conversions, providing allophone chains for an acoustic processor. The developed algorithms were evaluated with respect to the reference-class stationary Multiphone text-to-speech synthesis system. The test data demonstrates a high accuracy level: over 80% for word processing, over 98% for phoneme processing and over 96% for allophone processing (for Belarusian and Russian languages). The subsequent work will be devoted to the improvement of practical realization of the developed algorithms.