А. Ю. Станкевич, А. И. Богдевич (Гродно, ГрГУ)

АВТОМАТИЗАЦИЯ ПОИСКА МОРФОНОЛОГИЧЕСКИХ ЯВЛЕНИЙ ПО ДАННЫМ СЛОВАРЯ УКРАИНСКОГО ЯЗЫКА

Морфонологические единицы представляют собой образец языковых единиц, с одной стороны, характеризующихся высокой степенью зависимости от материальной репрезентации (поскольку являются в первую очередь фактами именно языка как субстанции), с другой же, в определенной степени независимых от языковой материи (поскольку степень вариативности собственно субстанционального наполнения таких единиц достаточно высока). В связи с этим проблема идентификации как самих морфонологических единиц, так и морфонологических явлений не относится к разряду легко автоматизируемых. Вместе с тем, в любом морфонологическом исследовании возможность автоматизации, несомненно, была бы в высшей степени востребованной, так как подобные исследования всегда связаны с изучением и обобщением огромного массива фактического материала.

Авторами предложен алгоритм автоматизированного поиска 2-х групп морфонологически обусловленных изменений субстантивной словоформы: 1) мена ударения и 2) замена, вставка, удаление символов относительно начальной формы. На первом этапе исследования и построения алгоритма идентификации к морфонологическим явлениям относим любые видоизменения первоначальной основы, не задаваясь вопросом о лингвистическом статусе таковых.

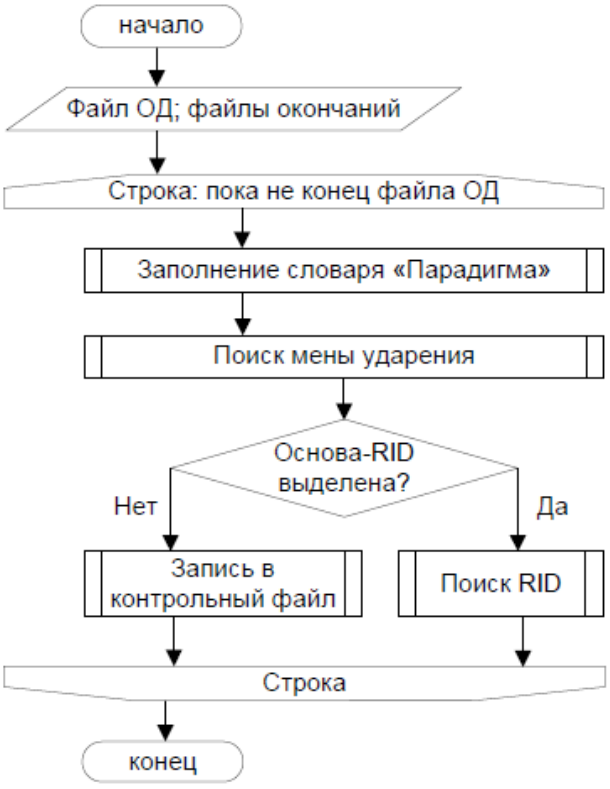

Рис. 1. **Укрупненная блок-схема алгоритма поиска МЯ**

Лингвистическая база данных субстантивного словоизменения украинского языка (далее ЛБД) [2], на материале которой апробирован алгоритм поиска морфонологических явлений (далее МЯ), содержит лексикографическое описание 2 300 субстантивных лексем, полученных методом сплошной выборки из онлайн-версии лексикографической системы «Словники України» [1].

Поиск МЯ осуществлялся средствами разработанного нами модуля, реализованного на Python 2.7. Файлы входа: 1) файл основных данных (далее — файл ОД), содержащий значения полей ЛБД [Код_лексемы], [Лексема], [Основа] и значения 7 полей с под-

парадигмами по падежам; 2) 2 файла окончаний: для единственного и для множественного числа (окончания даны с метками падежей). Файлы выхода: два файла со статистикой: stat.txt для словоформ с МЯ и stat_z для всех словоформ (в stat_z записывается и метка отсутствия МЯ).

Укрупненная блок-схема алгоритма поиска МЯ представлена на рис.

А. Функция «Заполнение словаря “Парадигма”«.

Функция возвращает файл словаря, где ключами являются: 1) строки вида «падеж_число» (значение по такому ключу — список словоформ, соответствующих этой составной грамматической характеристике или метка лакуны); 2) ключи Code, Lex, Base (значения — строки с кодом лексемы, лексемой, основой). Функция принимает строку файла ОД, где табуляцией отделены семь фрагментов, соответствующие подпарадигмам форм единственного и множественного числа каждого из падежей и еще три фрагмента, соответствующие коду лексемы, лексеме, основе. Каждый из 7 фрагментов с подпрадигмами разделяется по 1; ход предопределенных процессов прокомментирован ниже.

заданным в функции шаблонам и приписывается как значение к соответствующему ключу. Шаблоны заданы для 9 моделей (3 варианта представления единственного числа: {без вариантов, с вариантами, лакуна}, сочетающиеся с каждым из 3-х вариантов представления множественного числа: {без вариантов, с вариантами, лакуна}). Проверку реализации хотя бы по одному шаблону и проверку непересечения шаблонов обеспечивает вычисление контрольной суммы, равной числу реализованных на подпарадигме шаблонов: при верном разборе контрольная сумма должна быть равна 1. Заданное нами множество шаблонов обеспечивает верный разбор.

Б. Функция «Поиск мены ударения».

Функция принимает словарь «Парадигма» и обходит элементы с ключами вида «падеж_число», при этом в активном списке для каждой пары «основа — словоформа» ведется поиск одного из 2-х видов мены ударения: с основы на финаль (в файлы статистики записывается строка с меткой **b2f)** или с финали на основу (в файлы статистики записывается строка с меткой **f2b**); при отрицательном результате поиска в файл stat_z.txt записывается строка с меткой **NO**.

Пусть form — словоформа; formNA — словоформа без символа ударения; baseTrim — основа без знака мягкости основы и / или без конечного «*й*», причем в baseTrim нет отличных от мены ударения МЯ; baseTrimNA — baseTrim без символа ударения; основа-RID — основа formNA, где наблюдается хотя бы одно отличное от мены ударения МЯ (иначе говоря, основа-RID может быть получена из baseTrimNA путем выполнения одной и более операций из набора {**R**eplace, **I**nsert, **D**elete}); lex — лексема; FIN — подмножество очищенных от знака ударения окончаний, определенное сочетанием значений падежа и числа словоформы form; FIN_i — элемент такого подмножества.

Любая formNA может соответствовать одной из 2-х структур: «baseTrimNA + FIN_i» или «основа-RID + FIN_i».

Определение мены ударения для структуры первого типа тривиально. Для поиска мены ударения в структуре второго типа необходимо выделить **осно-**

ву-RID. Основа-RID выделяется путем **фильтрованного** перебора элементов подмножества FIN и проверки условия «formNA оканчивается на FIN_i»; при выполнении условия часть formNA без FIN_i записывается в словарь вариантов основы-RID с ключом, равным длине FIN_i; после завершения обхода FIN за основу-RID принимается вариант по наибольшему ключу (т.е. по самому длинному из потенциальных окончаний). Под **фильтрованным** перебором элементов подмножества FIN понимаем запрет на проверку условия «formNA оканчивается на FIN_{max}» для такого FIN_{max}, что «FIN_{max} оканчивается на FIN_{min}», если верно, что «baseTrimNA оканчивается на $FIN_{max} \setminus FIN_{min}$». Фильтры описаны в табл. 1.

Таблица 1

Фильтры для FIN_{max}

Определитель FIN	**FIN_{max} // FIN_{min}**	**Комментарий**
Sing Dat	*ому // у*	Завершение baseTrimNA: *-ом*. 9 реализаций: *автодро́м — автодро́му* и др.
Sing Dat	*ові, еві, єві // і*	Завершение baseTrimNA ∈ *{-ов, -їв, -ів, -ев, -єв}*. 2 реализации: *Айо́ва — Айо́ві* и др.
Sing Loc	*ові, еві, єві // і*	Завершение baseTrimNA ∈ {-ов, -їв, -ів, -ев, -єв}. Т.к. есть регулярное чередование о//{і, ї}, то исключения для основ на *-ов* верны и для основ на *-ів* и *-їв*. 53 реализации (максимум): *Агафо́нов-Агафо́нові* и др.
Plur Gen	*ей // й*	Завершение baseTrimNA: *-е*. 3 реализации: *але́я — але́й* и др.
Sing Acc	*ого // о*	Завершение baseTrimNA: *-ог*. На нашем материале: 0.
Sing Loc	*ому // у*	Завершение baseTrimNA: *-ом*. На нашем материале: 0.

Пусть finNA-RID — окончание формы структуры второго типа (часть formNA без основы-RID). В общем случае **для структуры второго типа** метки мены ударения определены следующими условиями:

– для метки **f2b**: «form: есть ударение И baseTrim: нет ударения И lex: есть ударение И finNA-RID непустое И form заканчивается на finNA-RID (т.е. окончание form безударное)»;

– для метки **b2f**: «form: есть ударение И baseTrim: есть ударение И finNA-RID непустое И form не заканчивается на finNA-RID (т.е. окончание form ударное)».

Функция начинает обработку всякой словоформы как структуры первого типа, при неуспехе — как структуры второго типа, при неуспехе данные передаются функции «Запись в контрольный файл».

В. Функция «Поиск RID».

Функция принимает значения baseTrimNA и основы-RID, для которых возвращается расшифровка редакционного предписания по расстоянию Левенштейна. Хотя существует официальная Python-реализация вычисления расстояния Левенштейна [3], по субъективным причинам мы обратились к представленной здесь [4] реализации. Адаптация выбранного кода свелась к расшифровке возвращаемых редакционных предписаний и записи соответствующей

строки в файл статистики. **Расшифровкой** называем преобразование редакционного предписания в записи вида «индекс операции:: фрагмент baseTrimNA // фрагмент основы-RID» для всех операций, кроме **M**atch; определена автоматическая нормализация расшифровки (например, не соответствующая описанию чередования расшифровка *R::ь//e*; нормализуется в *I::-//e*, расшифровка *R::e//ь* нормализуется в *D::e//-* и т.п.). Примеры расшифровки предписания даны в табл. 2.

Таблица 2

Расшифровка предписаний (для форм генетива плюралиса)

form	baseTrimNA	основа-RID	Предписание	Расшифровка
авіашкі́л	*авіашкол*	*авіашкіл*	[M M M M M M R M]	*R:: o//i*
англя́т	*англ*	*англят*	[M M M M M I I]	*I:: --//ят*
ава́рців	*аварец*	*аварц*	[M M M M D M]	*D::e//-*

Лингвистическую корректность результатов обеспечивает передача адаптированной функции Левенштейна не начальной формы и словоформы, а основы начальной формы (по значению ключа Base словаря «Парадигма») и основы-RID (полученной по функции «Поиск мены ударения» с обращением к словарю окончаний): очевидно, что при расшифровке предписаний для начальной формы и словоформы мы получим весьма экзотические по отношению к морфонологии факты. К примеру, расшифровка предписания для пары «*аварец-аварці*» ([M M M M R R]) даст чередование *R::ец//ці,* расшифровка предписания для пары «*алтаєц — алтайців*» ([M M M M R M I I]) даст два столь же некорректных чередования *(R:: є//й, I::--//ів*) и т.п.

Основные статистики выделенных МЯ приведены в табл. 4, 5.

Таблица 3

Частотный список МЯ

№	МЯ	Частота (%)	№	МЯ	Частота (%)	№	МЯ	Частота (%)
1.	b2f	603 (24.92)	9.	f2b	32 (1.32)	17.	R::ї//є	6(0.25)
2.	D::e//-	431 (17.81)	10.	I::--//ят	20 (0.83)	18.	R::o//i	2(0.08)
3.	R::i//o	410 (16.94)	11.	R::i//e	13 (0.54)	19.	I::-//д	1(0.04)
4.	R::к//ц	338 (13.97)	12.	R::х//с	9 (0.37)	20.	I::-//ж	1(0.04)
5.	I::-//o	182 (7.52)	13.	R::г//з	8(0.33)	21.	I::-//ш	1(0.04)
6.	D::o//-	126 (5.21)	14.	D::н//-	8(0.33)	22.	R::к//ч	1(0.04)
7.	D::є//-	112 (4.63)	15.	D::ин//--	7(0.29)	23.	I::-//з	1(0.04)
8.	I::-//e	102 (4.21)	16.	I::-//л	6(0.25)	ВСЕГО: 2420 (100)		

Таблица 4

Морфонологическая активность позиции в парадигме

№	Падеж, число	Всего форм	Форм с МЯ (%)	№	Падеж, число	Всего форм	Форм с МЯ (%)
1.	Gen, plur	1465	273 (18.63)	8.	Loc, plur	1468	102 (6.95)
2.	Acc, plur	1494	225 (15.06)	9.	Nom, plur	1468	102 (6.95)
3.	Loc, sing	3251	472 (14.52)	10.	Voc, plur	1468	102 (6.95)
4.	Dat, sing	3571	435 (12.18)	11.	Voc, sing	2472	159 (6.43)

№	Падеж, число	Всего форм	Форм с МЯ (%)	№	Падеж, число	Всего форм	Форм с МЯ (%)
5.	Gen, sing	2370	172 (7.26)	12.	Abl, sing	2368	106 (4.48)
6.	Abl, plur	1470	106 (7.21)	13.	Acc, sing	2362	61 (2.58)
7.	Dat, plur	1468	102 (6.95)	14.	Nom, sing	2357	3 (0.13)

В перспективе видим разработку алгоритма определения морфонологического / фонологического / фонетического статуса выявленных МЯ, что, в конечном счете, предоставит возможность создания программных средств автоматизированного анализа данных словарей для выявления и идентификации морфонологических явлений.

ЛИТЕРАТУРА

1. Словники України on-line [Электронный ресурс]. — Режим доступа: http://lcorp.ulif.org.ua/dictua. — Дата доступа: 10.03.2013.

2. Багдзевіч, А.І. Праектаванне лінгвістычнай базы дадзеных «Марфаналагічнае вывучэнне ўкраінскай мовы» // Карповские научные чтения. сб. науч. статей. — Минск, 2012. — С. 216–219.

3. Python-Levenshtein [Electronic resource]. — Mode of access: http://pypi.python.org/pypi/ python-Levenshtein. — Date of access: 10.03.2013.

4. Расстояние Левенштейна [Электронный ресурс]. — Режим доступа: http://muzhig.ru/levenstein-distance-python/. — Дата доступа: 10.03.2013.