УДК 81.33
DOI:10.25729/ESI.2026.41.1.015

Разработка программы «Калькулятор лингвиста»

Боровский Андрей Викторович, Мосоркин Федот Эдуардович

Байкальский государственный университет, Россия, Иркутск, *mosorkin@bk.ru*

Аннотация. В статье описывается разработка программного обеспечения для исследований в области историко-математической лингвистики, в которых используются мультиметрический подход и метод анализа иерархий. Программа реализована в виде десктопного приложения на языке программирования Python. Для разработки графического интерфейса применена библиотека PyQt5. Рассмотрены и реализованы актуальные математические методы для исследований в историко-математической лингвистике, такие, как: преобразования слов Долгопольского А.Д. в консонантные классы, различные метрики сходства слов (учитывающие количество одинаковых букв в двух словах (Рэтклиффа-Обершелпа или RO), количество букв в наибольшей общей подстроке (LCS), количество элементарных операций по совмещению слов (расстояние Левенштейна или L)). Новизна работы заключается в применении к анализу списка соответствий мультиметрического подхода и выстраивание рейтингов на основе метода анализа иерархий. Используя «Калькулятор лингвиста», можно выявлять скрытые лексические связи между топонимами и списками слов соответствий, а также проводить исследования происхождения топонимов. Программа апробирована на топонимах Иркутской области с утраченным смыслом и позволяет выявлять наиболее вероятные соответствия среди слов-кандидатов из различных языков: эвенкийский, бурятский, старорусский. Реализованы ввод топонима и слов-кандидатов, выбор модели преобразования слов, вывод и экспорт в Excel-файл отсортированных результатов по убыванию суммы метрик. Проведены процедуры верификации метода анализа иерархии в мультиметрии слов, для чего применены наборы слов, которые были специально изменены для проверки устойчивости метода к искажениям слов. В итоге исследование показало, что алгоритм устойчив к искажениям. При шуме 50% падение качества установления соответствий происходит постепенно. Устойчивость алгоритма к искажениям делает его пригодным для работы с реальными (в том числе с искаженными) топонимами. В будущем планируется добавить функции для количественной оценки заимствований в языках, чтобы расширить ее применение в историко-математической лингвистике для анализа языковых взаимодействий и реконструкции этимологии топонимики Иркутской области.

Ключевые слова: Историко-математическая лингвистика, разработка ПО, консонантные классы, парные метрики Рэтклиффа-Обершелпа, LCS, расстояние Левенштейна

Цитирование: Боровский А.В. Разработка программы «Калькулятор лингвиста» / А.В. Боровский, Ф.Э. Мосоркин // Информационные и математические технологии в науке и управлении, 2026. – № 1(41). – С. 199-206. – DOI:10.25729/ESI.2026.41.1.015.

Введение. В настоящее время в лингвистике рутинные задачи отнимают значительное количество времени из-за большого объема данных для работы. Чтобы справиться с подобными проблемами, применяют автоматизацию, путем разработки программ для обширных вычислений. Для этой цели было решено реализовать программу «Калькулятор лингвиста».

При работе над топонимами Иркутской области с утерянным смысловым значением авторы столкнулись со следующей проблемой. К конкретному топониму с непонятным для авторов смыслом подбирались похожие по произношению слова из словарей старорусского, эвенкийского, бурятского и других языков. Возникало отношение: топоним – список слов соответствий. Размерность списка могла достигать 10-20 и более. Возникала задача, как математическим путем выстроить рейтинг слов соответствий в найденном списке. Применение какой-то одной парной метрики, описывающей близость слов, не давало необходимого эффекта. В связи с этим авторы предлагают для решения задачи использовать мультиметрический подход и метод анализа иерархий из области системного анализа.

Цель работы – разработать программу «Калькулятор лингвиста», предназначенную для вычисления нескольких парных метрик и выстраивания рейтингов в списках слов соответствий к топониму.

Для достижения цели были поставлены следующие задачи:

- рассмотреть методы и алгоритмы, применяемые при сравнении слов;
- выявить требования к программе;
- реализовать функционал программы;
- верифицировать методы, примененные в программе.

Методологическое обоснование. Языки имеют свойство со временем меняться, смыслы и звучание старых слов забываются и перестают использоваться в речи. Поэтому в лингвистике используют различную кодировку звучания слов. Например, есть методы сегментации (фонемная и слоговая), фонетические транскрипции (IPA, ARPAbet) и т.п. Однако программа нацелена для работы в области историко-математической лингвистики, поэтому для преобразования слов в программе использованы консонантные классы А. Д. Долгопольского (табл. 1) [1]. В самом программном приложении выделены два типа преобразований:

- Д0: исходные слова без преобразований;
- Д1: преобразование гласных в класс Н и согласных в соответствующий консонантный класс.

Таблица 1. Консонантные классы для русского языка

№	Класс согласных	Буквы русского языка
1	P-класс	П, Б, В, Ф
2	T-класс	Т, Д
3	S-класс	С, З, Ц, Ч, Ш, Щ, Ч
4	M-класс	М
5	N-класс	Н
6	R-класс	Р, Л
7	K-класс	К, Г, Х
8	H-класс (нулевой класс)	Все гласные, включая Ё, Й, Ю, Я

Чтобы оценить сходство слов, были использованы три метрики:

– Рэтклиффа-Обершелпа [4], основанная на гештальтном подходе к сопоставлению слов. Фиксирует фрагментарные совпадения даже при значительных различиях слов. По факту эта метрика указывает долю общих букв без учета порядка их следования в двух словах. Если слова совпадают, то метрика РО равна 1. Если в словах переставлены буквы или слоги, то метрика РО также равна 1. Если слова отличаются хотя бы одной буквой, то метрика РО <1. Если все буквы в словах разные, то метрика РО равна 0. Метрика РО позволяет существенно ограничить поле для поиска соответствий.

– Наибольшая общая подстрока (Longest Common Subsequence) [5]. Данная метрика выявляет максимально длинную последовательность, общую для двух слов, что информирует о степени их сходства при сохранении порядка элементов. Метрика LCS позволяет забраковать случаи, когда буквенный состав слов близкий, но буквы перемешаны.

– Расстояние Левенштейна [6], равно минимальному числу элементарных операций (вставка, удаление, замена символа), необходимых для преобразования одной строки в другую. Для его реализации в программе был использован алгоритм Вагнера-Фишера [7]. При исторической эволюции слова меняются постепенно, по одной операции на определённом отрезке времени. Поэтому большие значения метрики L указывают на значительную временную удаленность таких изменений.

Для ранжирования списка совпадений в программе используются два метода: мультиметрический и метод анализа иерархий. В мультиметрическом методе используем три метрики: PO, LCS, L. В методе анализа иерархий вводятся объективные критерии для оценки совпадений. В нашем случае выбираем один топоним и для него имеем совокупность слов-совпадений. Для каждого слова из списка совпадений рассчитываются три метрики, другими словами, три критерия. Далее суммируем их и по величине суммы ранжируем список совпадений.

Первые две метрики рассчитываются по формуле

$$\rho = \frac{2K(a,b)}{|a|+|b|}, \quad (1)$$

метрика L рассчитываются по формуле

$$\rho = 1 - \frac{2K(a,b)}{|a|+|b|}, \quad (2)$$

где a – первое слово, b – второе слово, $|a|, |b|$ – количество букв в первом и втором словах, $K(a,b)$ – количество общих букв в каждом слове для метрики PO, количество букв в максимальной общей подпоследовательности с учетом разрывов и порядка следования букв для метрики LCS, количество элементарных операций, переводящих одно слово в другое для метрики L.

Инструменты разработки и этап проектирования. Предлагаемая программа написана на языке программирования Python, как наиболее подходящем для работы с данными. Интерфейс реализован с помощью библиотеки PyQt5 [2-3] в виде десктопного оконного приложения. Чтобы экспортировать результаты в формате Excel-файла, используется библиотека openpyxl [8]. Для составления графиков была использована библиотека matplotlib [9].

Программа «Калькулятор лингвиста» встроена в виде отдельного модуля в программу «Помощник лингвиста» [10], которая разработана для сравнения двух лингвистических списков слов.

Для разработки архитектуры модуля «Калькулятор лингвиста» была сформирована диаграмма прецедентов (Use Case), приведенная на рис. 1. В ней представлены взаимодействия пользователя с программой, выявляющие основные пользовательские требования и сценарий использования приложения.

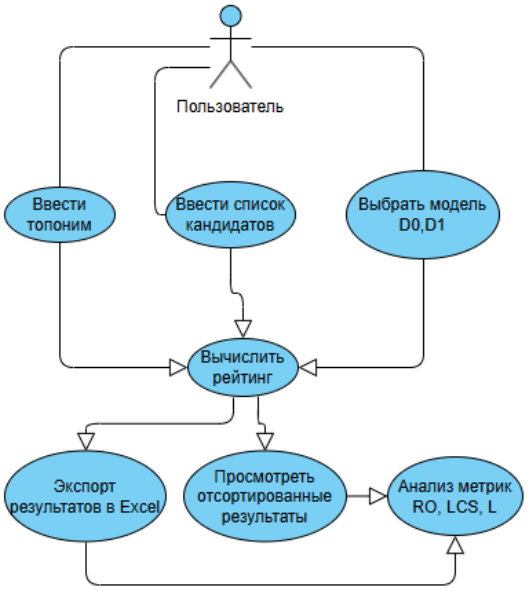

Рис. 1. Use Case-диаграмма прецедентов

Исходя из приведенной диаграммы, составлены следующие требования к модулю:

– простой ввод данных – возможность ручного ввода топонима и списка кандидатов;
– гибкий выбор модели – Д0 или Д1;
– наглядный просмотр результатов, отсортированных по убыванию суммы метрик;
– экспорт результатов – сохранение таблицы с кандидатами и значениями метрик в Excel.

На рис. 2 представлен интерфейс модуля «Калькулятор лингвиста».

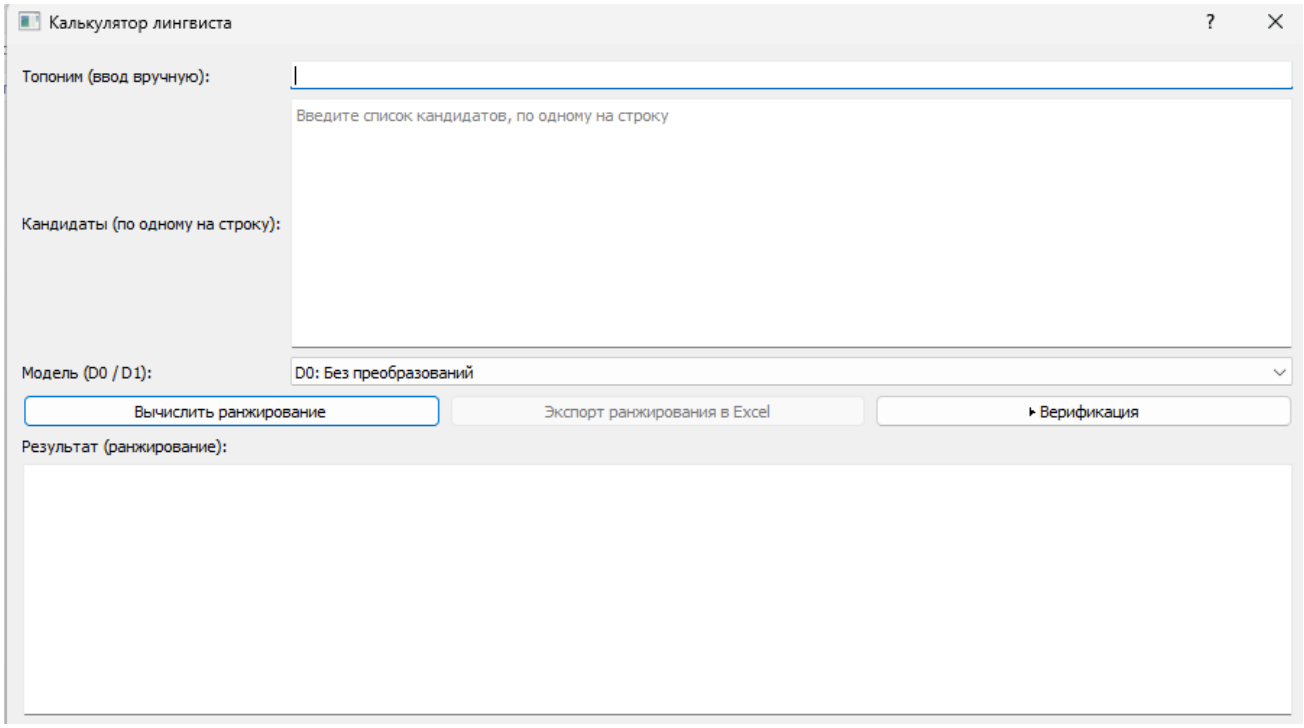

Рис. 2. Интерфейс модуля «Калькулятор лингвиста»

Алгоритм работы: вводим топоним, затем построчно вставляем список похожих слов. При необходимости можно выбрать модель преобразования слов Д0 или Д1. Нажимаем «Вычислить рейтинг» и получаем результат. Рассчитываются метрики RO, LCS, L и их сумма, по которым можно определить наиболее подходящие кандидаты для топонима.

Для примера возьмем топоним «барда». Набор кандидатов следующий: бард, барда, бурда, брага, борода, брада, бразда, борозда, бурят, брат, брады, рада, нард, обада, обида. Вычислим рейтинг с моделью преобразования Д0. Нажимаем «вычислить рейтинг» и модуль выводит результаты (рис. 3.).

Результат (ранжирование):

Топоним: Барда
Модель: D0

#	Кандидат	RO	LCS2	LEV	SUM
1.	барда	1.000	1.000	1.000	3.000
2.	бард	0.889	0.889	0.778	2.556
3.	бурда	0.800	0.800	0.800	2.400
4.	брада	1.000	0.800	0.600	2.400
5.	бразда	0.909	0.727	0.636	2.273
6.	обада	0.800	0.800	0.600	2.200
7.	рада	0.889	0.667	0.556	2.111
8.	борода	0.727	0.727	0.636	2.091
9.	нард	0.667	0.667	0.556	1.889
10.	борозда	0.667	0.667	0.500	1.833
11.	брага	0.800	0.600	0.400	1.800
12.	брады	0.800	0.600	0.400	1.800
13.	брат	0.667	0.667	0.333	1.667
14.	обида	0.600	0.600	0.400	1.600
15.	бурят	0.400	0.400	0.400	1.200

Рис. 3. Результат работы модуля «Калькулятор лингвиста» на примере топонима «барда»

Верификация. Чтобы проверить устойчивость иерархического метода к искажениям слов-кандидатов, создадим «испорченные» наборы изначальных кандидатов. В таблице 2 описываются операции для преобразования. Наборы будут искажены на определенный процент (в нашем случае 10%, 20%, 30%, 40%, 50%). Все перечисленные операции будут комбинироваться при формировании «испорченных» кандидатов. Для каждого уровня шума будет 5 искаженных наборов.

Таблица 2. Операции для искажения кандидатов

Операция	Что делает	Пример	Лингвистический смысл
Замена буквы	Заменяет случайные буквы другими (обычно с сохранением длины)	Москва → Мшсква	Имитация ошибок распознавания
Удаление буквы	Удаляет одну или несколько букв в случайных местах	Самара → Смара	Потеря символов при сканировании
Вставка буквы	Добавляет лишние буквы	Казань → Казаньк	Имитация артефактов, ошибки автоматических конвертаций
Перестановка букв	Меняет местами буквы	Минск → Мискн	Типовая опечатка

В таблице 3 представлен пример искаженных слов кандидатов для топонима «Барда» при уровне шума 20%. Исходный набор слов кандидатов тот же.

Таблица 3. Примеры искаженных слов кандидатов

№	Кандидат	№	Кандидат	№	Кандидат	№	Кандидат	№	Кандидат
1	бара	1	баря	1	кард	1	бард	1	брад
2	барда	2	бара	2	ураад	2	ъарда	2	барды
3	буида	3	урв	3	бруда	3	урда	3	аурда
4	рбаба	4	брыга	4	брага	4	брбга	4	браг
5	бородч	5	бурода	5	борода	5	борода	5	борода
6	брада	6	брад	6	брада	6	Бада	6	брада
7	бразд	7	бразда	7	бразда	7	бразда	7	бразда
8	корозйа	8	ороздъ	8	оброзда	8	боодд	8	бороз
9	бурят	9	шурят	9	бубят	9	руяё	9	бурзт
10	браб	10	бпат	10	барж	10	брт	10	брат
11	ради	11	барды	11	брвгы	11	брады	11	мшады
12	радв	12	раа	12	юа	12	рада	12	жад
13	вард	13	над	13	над	13	урд	13	наро
14	баюа	14	чвада	14	обааа	14	биада	14	бада
15	обида	15	обдиа	15	обчда	15	онида	15	обдда

Результаты верификации иерархического метода на примере топонима «Барда» представлены в таблице 4 и на графике (рис. 4).

Таблица 4. Результаты верификации иерархического метода

Уровень шума	Средняя SUM без искажений	Средняя SUM с искажениями	Разница SUM (delta)	Относительное падение (delta / средняя SUM без искажений)
0	2,054613	2,054613	0	0
10		1,941892	0,112721	0,054862
20		1,701883	0,35273	0,171677
30		1,64276	0,411853	0,200453
40		1,464633	0,58998	0,287149
50		1,288574	0,766038	0,372838

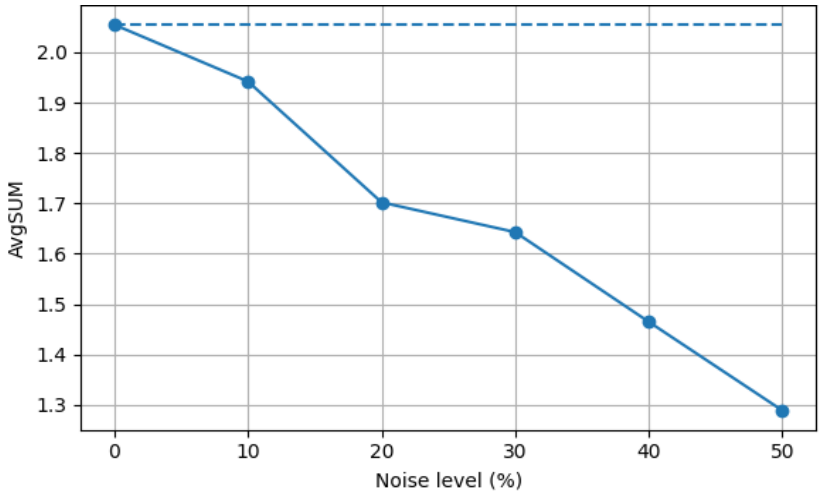

Рис. 4. График изменения средней суммы в зависимости от уровня шума

Результаты исследования следующие:

- При 10% шума падение средней суммы составляет всего 5,5% – метод устойчив к небольшим ошибкам;
- 30% шума – падение 20% – метод показал неплохой результат;
- 40% шума – падение 28% – метод показал неплохой результат;

Даже при 50% шума система не разваливается полностью, так как SUM> 1.28. Падение средней суммы – 37%.

Заключение. Разработана и апробирована программа «Калькулятор лингвиста», которая является модулем программы «Помощник лингвиста». Новизной выполненной работы является применение к анализу списка соответствий мультиметрического подхода и выстраивание рейтингов на основе метода анализа иерархий.

Реализованы следующие функции модуля:

- различные виды преобразования слов в консонантные классы;
- вычисление трех различных метрик, описывающих близость слов;
- выстраивание рейтинга слов соответствий на основе уменьшения суммы метрик;
- экспорт результатов вычислений в виде Excel-файла.

Выполнена верификация иерархического метода на устойчивость к искажениям. Алгоритм хорошо переносит шумы до 40%, даже при 50% качество установления соответствий снижается постепенно, без критического обвала.

Таким образом, данная программа поможет лингвистам в исследованиях, например, при нахождении скрытых лексических связей между двумя списками слов, что, в свою очередь, находит применение в изучении топонимики. Программа будет дополнена функцией определения заимствований в языках [11].

Список источников

1. Боровский А.В. Изучение связи между русским и бурятским языками методом матрицы мер близости между консонантными классами слов / А.В. Боровский, В.В. Братищенко, Е.Е. Раковская // System Analysis & Mathematical Modeling, 2023. – Т. 5. – № 1. – С. 19–33.
2. Библиотека PyQt5 – URL: https://doc.qt.io/qtforpython-5/ (дата обращения: 05.09.2025).
3. Язык программирования Питон – URL: https://docs.python.org/3.12/ (дата обращения: 05.09.2025).
4. Рэтклифф Д.В. Сопоставление образцов: гештальтный подход / Д.В. Рэтклифф, Д.Э. Меценер // Dr. Dobb's Journal, 1988. – № 46. – С. 46.
5. Гасфилд Д. Алгоритмы на строках, деревьях и последовательностях: информатика и вычислительная биология / Д. Гасфилд. – Кембридж: Cambridge University Press, 1997. – С. 230–235.
6. Левенштейн В.И. Двоичные коды, способные исправлять удаления, вставки и обратные операции / В.И. Левенштейн // Советская физика – Доклады, 1966. – Т. 10, № 8. – С. 707–710.
7. Лещенко А. В. Практическое применение алгоритмов нечеткого поиска. Сборник научных трудов НГТУ. – 2018. – № 3–4 (93).
8. Библиотека openpyxl – URL: https://openpyxl.readthedocs.io/ (дата обращения: 05.09.2025).
9. Matplotlib: Visualization with Python – URL: https://matplotlib.org/stable/ (дата обращения: 05.09.2025).
10. Мосоркин Ф.Э. Разработка программы «Помощник лингвиста» // System Analysis & Mathematical Modeling, 2025. – Т. 7, № 3. – С. 419–427. – DOI: 10.17150/2713-1734.2025.7(3).419-427. – EDN: ETGMWD.
11. Боровский А.В. Количественное определение заимствований в языке / А.В. Боровский, Е.Е. Раковская, Ф.Э. Мосоркин // System Analysis & Mathematical Modeling, 2025. – Т. 7, № 3. – С. 333–345. – DOI: 10.17150/2713-1734.2025.7(3).333-345. – EDN: KZPABO.

Боровский Андрей Викторович. *Доктор физико-математических наук, профессор, кафедра математических методов и цифровых технологий, Байкальский государственный университет, г. Иркутск, Российская Федерация. AuthorID: 22229, SPIN: 7243–8706. ORCID:0000-0003-2119-1072, andrei-borovskii@mail.ru.*

Мосоркин Федот Эдуардович. *Аспирант, кафедра математических методов и цифровых технологий, Байкальский государственный университет, г. Иркутск, Российская Федерация. AuthorID: 1333242, SPIN: 7830-4843, mosorkin@bk.ru.*

Вклад авторов

Боровскому А.В. принадлежит идея метода; Мосоркину Ф.Э. принадлежит разработка программного обеспечения и верификация метода

UDC 81.33
DOI:10.25729/ESI.2026.41.1.015

Development of the "Linguist's Calculator" program

Andrei V. Borovsky, Fedot E. Mosorkin

Baikal State University, Russia, Irkutsk, *mosorkin@bk.ru*

Abstract. This article describes the development of software for research in historical and mathematical linguistics that utilizes a multimetric approach and the analytic hierarchy process. The program is implemented as a desktop application in the Python programming language. The PyQt5 library was used to develop the graphical interface. Relevant mathematical methods for research in historical and mathematical linguistics are considered and implemented, including: transformations of A.D. Dolgopolsky's words into consonant classes, various word similarity metrics (taking into account the number of identical letters in two words (Ratcliff-Obershelp or RO), the number of letters in the longest common substring (LCS), and the number of elementary operations for combining words (Levenshtein distance or L)). The novelty of this work lies in the application of a multimetric approach to

the analysis of the list of correspondences and the construction of rankings based on the analytic hierarchy process. The "Linguist's Calculator" allows one to identify hidden lexical relationships between toponyms and lists of corresponding words, as well as conduct research into the origins of toponyms. The program has been tested on toponyms of the Irkutsk region with lost meanings and identifies the most likely matches among candidate words from various languages, including Evenki, Buryat, and Old Russian. The program supports input of a toponym and candidate words, selection of a word transformation model, and output and export of sorted results in descending order of metric sum to an Excel file. Verification procedures were conducted for the hierarchy analysis method in word multimetrics, using word sets that were specifically modified to test the method's robustness to word distortions. The study demonstrated that the algorithm is robust to distortions. With 50% noise, the quality of matching gradually declines. The algorithm's robustness to distortions makes it suitable for working with real (including distorted) toponyms. In the future, we plan to add functionality for quantitatively assessing borrowings in languages to expand its application in historical and mathematical linguistics for analyzing linguistic interactions and reconstructing the etymology of toponymy in the Irkutsk region.

Keywords: Historical and mathematical linguistics, software development, consonant classes, Ratcliffe-Obershelp pairwise metrics, LCS, Levenshtein distance

References

1. Borovsky A.V., Bratishchenko V.V., Rakovskaya E.E. Izucheniye svyazi mezhdu russkim i buryatskim yazykami metodom matritsy mer blizosti mezhdu konsonantnymi klassami slov [Study of the relationship between Russian and Buryat languages using the method of a proximity measure matrix between consonantal word classes]. System Analysis & Mathematical Modeling, 2023, vol. 5, no. 1, pp. 19–33.
2. PyQt5 Library. Available at: https://doc.qt.io/qtforpython-5/ (accessed: 09/05/2025).
3. Python Programming Language. Available at: https://docs.python.org/3.12/ (accessed: 09/05/2025).
4. Ratcliff J.W., Metzener D.E. Pattern Matching: The Gestalt Approach. Dr. Dobb's Journal, 1988, no. 46, p. 46.
5. Gusfield D. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge, Cambridge University Press, 1997, pp. 230–235.
6. Levenshtein V.I. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics – Doklady, 1966, vol. 10, no. 8, pp. 707–710.
7. Leshchenko A.V. Prakticheskoye primeneniye algoritmov nechetkogo poiska [Practical application of fuzzy search algorithms]. Sbornik nauchnykh trudov NGTU [Collection of Scientific Papers of NSTU], 2018, no. 3–4 (93).
8. openpyxl Library. Available at: https://openpyxl.readthedocs.io/ (accessed: 09.05.2025).
9. Matplotlib: Visualization with Python. Available at: https://matplotlib.org/stable/ (accessed: 09.05.2025).
10. Mosorkin F.E. Razrabotka programmy «Pomoshchnik lingvista» [Development of the "Linguist's Assistant" program]. System Analysis & Mathematical Modeling, 2025, vol. 7, no. 3, pp. 419–427, DOI: 10.17150/2713-1734.2025.7(3).419-427, EDN: ETGMWD.
11. Borovsky A.V., Rakovskaya E.E., Mosorkin F.E. Kolichestvennoye opredeleniye zaimstvovaniy v yazyke [Quantitative determination of borrowings in language]. System Analysis & Mathematical Modeling, 2025, vol. 7, no. 3, pp. 333–345, DOI: 10.17150/2713-1734.2025.7(3).333-345, EDN: KZPABO.

Borovsky Andrei Viktorovich. *D.Sc. in Physics and Mathematics, Professor, Department of Mathematical Methods and Digital Technologies, Baikal State University, Irkutsk, Russian Federation, AuthorID: 22229, SPIN: 7243–8706. ORCID:0000-0003-2119-1072, andrei-borovskii@mail.ru,*

Mosorkin Fedot Eduardovich. *PhD Student, Department of Mathematical Methods and Digital Technologies, Baikal State University, Irkutsk, Russian Federation. AuthorID: 1333242, SPIN: 7830-4843, mosorkin@bk.ru.*

Contribution of the Authors

Borovsky A.V. belongs to the idea of the method; Mosorkin F.E. belongs to the development of software and method verification.

Статья поступила в редакцию 07.11.2025; одобрена после рецензирования 17.11.2025; принята к публикации 21.02.2026.

The article was submitted 11/07/2025; approved after reviewing 11/17/2025; accepted for publication 02/21/2026.