7. **Skripak I. A.** Jazykovoe vyrazhenie jekspressivnosti kak sposoba rechevogo vozdejstvija v sovremennom nauchnom diskurse: na materiale statej lingvisticheskogo profilja na russkom i anglijskom jazykah. Phd thesis. Stavropol', 2008. 199 p. (in Russian).
8. **Smirnova L. N.** *Kurs anglijskogo jazyka dlja nauchnyh rabotnikov.* Leningrad: Nauka, 1971, 330 p. (in Russian).
9. **Smirnova L. N.** *Scientific English. Anglijskij jazyk dlja nauchnyh rabotnikov. Kurs dlja nachinajushhih.* Leningrad: Nauka, 1980, 245 p. (in Russian).
10. **Cvilling M. Ja.** *Nauchnaja literatura: jazyk, stil', zhanry*, Moscow: Nauka, 1985, 336 p. (in Russian).
11. **Academic** Phrasebank, available at: http://www.phrasebank.manchester.ac.uk/ (date of access: 20.01.2015).
12. **Anthony L.** Characteristic features of research article titles in computer science, *IEEE Transactions on Professional Communication*, 2001, no. 44 (3), pp. 187—194.
13. **Guse J.** *Communicative Activities for EAP.* Cambridge, Cambridge University Press, 2013, 322 p.
14. **Hamp-Lyons L., Heasley B.** *Study writing.* Cambridge, Cambridge University Press, 2013, 213 p.
15. **IEEE** — The world's largest professional association for the advancement of technology, URL: https://www.ieee.org/index.html (date of access: 05.06.2015).
16. **Jacobs P. S., Krupka G. R., Rau L. F.** Lexico-Semantic Pattern Matching as a Companion to Parsing in Text Understanding, *Workshop on Speech and Natural Language colocated with the 6th Human Language Technology Conference*, 1991, pp. 337—341.
17. **Jordan R. R.** English for academic purposes, Cambridge, Cambridge University Press, 2012, 404 p.
18. **Khosmood F., Levinson R. A.** Automatic natural language style classification and transformation, *BCS Corpus Profiling Workshop*, 2008, available at: https://style.soe.ucsc.edu/sites/default/files/CP08-KL-camera.pdf (date of access: 08.02.2017).
19. **Laurence** Anthony's AntConc, available at: http://www.laurenceanthony.net/software/antconc/ (date of access: 08.02.2017).
20. **Luyckx K., Daelemans W.** Shallow text analysis and machine learning for authorship attribution, *Computational Linguistics in the Netherlands 2004: selected papers from the Fifteenth CLIN Meeting*, van der Wouden T. [Ed.], e. a., Utrecht, LOT, 2005, pp. 149—160.
21. **Scolz T., Conrad S.** Style Analysis of Academic Writing, *Natural Language Pocessing and Information Systems: 16th International Conference on Appications of Natural Language to Information Systems*, Proceedings. NLDB 2011, Alicante, Spain, June 28—30, 2011, pp. 246—249.
22. **Siepmann D.** *Writing in English*: A Guide for Advanced Learners / J. D. Gallagher, M. Hannay; J. L. Mackenzie. — UTB, 2011, 469 p.
23. **Strinyuk S. A., Shuchalova Y., Lanin V.** Academic Papers Evaluation Software, *Application of Information and Communication Technologies (AICT), 2015 9th International Conference (14—16 Oct. 2015. Rostov-on-Don)*: IEEE, 2015, pp. 506—510.
24. **Swales J. M., Feak C. B.** *Academic writing for graduate students. Essential tasks and skills.* The University of Michigan, 2014, 418 p.
25. **TÜPH.** Nauchnyy portal, available at: https://kaemus.psych.ut.ee/&lang=Rus (date of access: 15.05.2017) (in Russian).
26. **Turabian K.** A Manual For Writers of Term Papers, Theses and Dissertations. — 7th ed. — The University of Chicago Press, 2007, 470 p.
27. **Wallwork A.** *English for academic research: vocabulary exercises*, London, Springer, 2013, 193 p.

УДК 004.421.6 DOI: 10.17587/it.24.523-528

Н. И. Лиманова, д-р техн. наук, проф., e-mail: Nataliya.I.Limanova@gmail.com,
М. Н. Седов, аспирант, e-mail: SedovMN@inbox.ru,
Поволжский государственный университет телекоммуникаций и информатики, г. Самара

Алгоритм нечеткого поиска реквизитов физических лиц в базах данных на основе метрики Левенштейна

При передаче данных от одного учреждения к другому возникает проблема персональной идентификации физических лиц, у которых частично или полностью не совпадают реквизиты. В работе представлен алгоритм нечеткого поиска, использующий модифицированную метрику Левенштейна, позволяющий выполнять поиск физических лиц в базе данных на основе нечеткого сравнения. Алгоритм реализован на языке PL-SQL в СУБД Oracle 11g.

Ключевые слова: *межведомственный информационный обмен, нечеткое сравнение, поиск персональных данных, функция интеллектуального сравнения, персональный идентификационный номер (ПИН)*

Введение

В процессе обработки информации о физических лицах в базах данных для удобства обработки каждому набору реквизитов физических лиц (таких как ФИО, адрес, номера паспорта, СНИЛС и т. п.) присваивается так называемый персональный идентификационный номер (ПИН). В случае обработки или передачи данных о физическом лице вся привязка осуществляется именно к этому ПИНу. При осуществлении обмена информацией о физических лицах между различными учреждениями возникает проблема сопоставления реквизитов из одной базы данных реквизитам в другой. Если проводить данное сопоставление методом простого сравнения реквизитов (метод прямого сравнения), то в случае ошибочных данных,

полученных, например, при ошибках ввода или ином искажении реквизитов, такие данные найдены не будут. Для решения этой проблемы применяют так называемый нечеткий поиск, при котором выполняется сопоставление информации заданному образцу поиска или близкому к этому образцу значению.

Соответственно, для однозначной привязки реквизитов физического лица из базы-источника необходимо выполнять нечеткий поиск реквизитов в базе-приемнике, который должен учитывать множество факторов: потенциальные ошибки при ручном вводе, отсутствующие или устаревшие реквизиты и т. п. Такой поиск целесообразно реализовать в виде алгоритма нечеткого поиска и основанного на нем специализированного программного обеспечения [1].

Известные алгоритмы нечеткого поиска строк

Рассмотрим существующие алгоритмы нечеткого поиска и проанализируем их [2]. Начнем с разработанного Робертом Расселом (Robert C. Russel) и Маргарет Кинг Оделл (Margaret King Odell) алгоритма Soundex [3]. Это один из алгоритмов сравнения двух строк по их звучанию. Он устанавливает одинаковый индекс для строк, имеющих схожее звучание в языке согласно заданной таблице схожих по звучанию символов и их сочетаний. Однако он имеет существенный недостаток: этот алгоритм привязан к языку, на котором написаны анализируемые строки. Данный алгоритм используется сейчас в основном в англоговорящей среде, для которой подобная таблица уже существует.

Следующий из рассматриваемых алгоритмов — алгоритм расширения выборки [4] — часто применяется в системах проверки орфографии. Он основан на сведении задачи о нечетком поиске к задаче о точном поиске. Этот алгоритм подразумевает построение наиболее вероятных "неправильных" вариантов поискового шаблона. Основное достоинство алгоритма заключается в легкости его модификации для генерации "ошибочных" вариантов по произвольным правилам. У алгоритма есть и недостатки, главный из которых — большое число проверок для слов существенной длины, поскольку из них можно получить много "ошибочных" слов.

Широко известен алгоритм на основе кода Хэмминга, который применяется при кодировании и декодировании данных. Линейные коды, как правило, хорошо справляются с редкими и большими опечатками. Однако их эффективность при сравнении слов с частыми, но небольшими ошибками достаточно низкая. В данном алгоритме также присутствуют дополнительные затраты на кодирование информации.

Следующий из рассматриваемых алгоритмов не совсем подходит под поставленную задачу, но для полноты картины не упомянуть о нем все же нельзя. Это алгоритм, использующий триангуляционные деревья, которые позволяют индексировать множества произвольной структуры при условии, что на них задана метрика. Существует довольно много различных модификаций данного алгоритма, но все они не слишком эффективны в случае текстового поиска и чаще используются в базе данных изображений или других сложных объектов.

Алгоритм Bitap (также известный как Shift-Or или Baeza-Yates-Gonnet) и различные его модификации наиболее часто используют для нечеткого поиска без индексации [5]. Впервые идею этого алгоритма предложили Ricardo Baeza-Yates и Gaston Gonnet, опубликовав соответствующую статью в 1992 г. Оригинальная версия алгоритма имеет дело только с заменами символов и фактически вычисляет расстояние Хемминга. Но немного позже Sun Wu и Udi Manber предложили модификацию этого алгоритма для вычисления расстояния Левенштейна, т. е. привнесли поддержку вставок и удалений и разработали на его основе первую версию утилиты Unix — agrep. Высокая скорость работы этого алгоритма обеспечивается за счет битового параллелизма вычислений — за одну операцию возможно провести вычисления над 32 и более битами одновременно. При этом тривиальная реализация поддерживает поиск слов длиной не более 32 символов. Использование типов больших размерностей замедляет работу алгоритма.

Рассмотрим далее алгоритм Вагнера—Фишера [6], который позволяет для двух строк найти расстояние Левенштейна — минимальное число операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую. Данный алгоритм имеет ряд значительных преимуществ перед всеми описанными выше, а именно: относительно невысокую сложность реализации; возможность качественного сравнения схожести более чем двух строк; несколько вариантов реализации, которые можно использовать в зависимости от конфигурации системы; универсальность для всевозможных алфавитов. Также у данного алгоритма существует одна интересная модифи-

кация, которая позволяет находить расстояние Дамерау—Левенштейна [7]. В нем к операциям вставки, удаления и замены символов, определенных в расстоянии Левенштейна, добавлена операция транспозиции (перестановки) символов. Ф. Дамерау показал, что 80 % ошибок при наборе текста человеком являются транспозициями. Вследствие всех перечисленных выше плюсов именно метрика Левенштейна была выбрана для основы рассматриваемого алгоритма нечеткого поиска реквизитов физических лиц в базах данных.

Математическая модель

Рассмотрим общую метрику Левенштейна, которая поддерживает три операции со строкой: вставки, замены и удаления символа, причем все три операции имеют одинаковый вес [8, 9]. Для дальнейшей работы была построена лингвистическая переменная "схожесть строк". Решено выделить следующие термы: "строки совпадают", "строки почти совпадают", "строки похожи", "строки и похожи, и непохожи одновременно", "строки не похожи".

В результате анализа функций принадлежности лингвистических термов возникла необходимость модификации метода вычисления метрики Левенштейна. Потребовалось модифицировать метрику таким образом, чтобы расстояние между строками зависело в том числе и от длины сравниваемых строк.

Теорема. Обозначим как $p(s_1, s_2)$ метрику Левенштейна, а $\|s_i\|$ длину строки s_i. Тогда функция

$$r(s_1, s_2) = \frac{p(s_1, s_2)}{\max\{\|s_1\|, \|s_2\|\}} \qquad (1)$$

является метрикой.

Доказательство. Поскольку $p(s_1, s_2)$ — метрика, то имеем

$$p(s_1, s_2) \geqslant 0,$$
$$p(s_1, s_2) = p(s_2, s_1),$$
$$p(s_1, s_2) + p(s_2, s_3) \geqslant p(s_1, s_3)$$

для любых строк s_1, s_2 и s_3.

Учитывая эти соотношения и равенство (1), приходим к выводу, что $r(s_1, s_2)$ удовлетворяет первым двум аксиомам, определяющим метрику. Остается доказать, что для любых строк s_1, s_2 и s_3 функция $r(s_1, s_2)$ удовлетворяет неравенству треугольника:

$$r(s_1, s_2) + r(s_2, s_3) \geqslant r(s_1, s_3).$$

Запишем это неравенство в виде:

$$\frac{p(s_1, s_2)}{\max\{\|s_1\|, \|s_2\|\}} + \frac{p(s_2, s_3)}{\max\{\|s_2\|, \|s_3\|\}} - \frac{p(s_1, s_3)}{\max\{\|s_1\|, \|s_3\|\}} \geqslant 0.$$

Возможны следующие случаи:

1. $\|s_1\| \leqslant \|s_2\| \leqslant \|s_3\|$.
2. $\|s_2\| \leqslant \|s_3\| \leqslant \|s_1\|$.
3. $\|s_3\| \leqslant \|s_1\| \leqslant \|s_2\|$.
4. $\|s_2\| \leqslant \|s_1\| \leqslant \|s_3\|$.
5. $\|s_1\| \leqslant \|s_3\| \leqslant \|s_2\|$.
6. $\|s_3\| \leqslant \|s_2\| \leqslant \|s_1\|$.

Рассмотрим первый случай. Имеем:

$$\frac{p(s_1, s_2)}{\max\{\|s_1\|, \|s_2\|\}} + \frac{p(s_2, s_3)}{\max\{\|s_2\|, \|s_3\|\}} -$$
$$- \frac{p(s_1, s_3)}{\max\{\|s_1\|, \|s_3\|\}} = \frac{p(s_1, s_2)}{\|s_2\|} + \frac{p(s_2, s_3)}{\|s_3\|} - \frac{p(s_1, s_3)}{\|s_3\|} \geqslant$$
$$\geqslant \frac{1}{\|s_3\|}(p(s_1, s_2) + p(s_2, s_3) - p(s_1, s_3)) \geqslant 0.$$

Таким образом, для первого случая неравенство треугольника выполняется. Поскольку второй случай аналогичен первому, на основании подобных выкладок делаем вывод, что для второго случая неравенство треугольника также выполняется.

Перейдем к рассмотрению третьего случая. Итак, в третьем случае имеем:

$$r(s_1, s_2) + r(s_2, s_3) - r(s_1, s_3) =$$
$$= \frac{1}{\|s_2\|}(r(s_1, s_2) + r(s_2, s_3)) - \frac{1}{\|s_1\|} r(s_1, s_3)). \qquad (2)$$

Рассмотрим вопрос о том, когда достигается минимум функции, находящейся в правой части этого равенства. Понятно, что если выражение $r(s_1, s_2) + r(s_2, s_3)$ достигает минимума, а $r(s_1, s_3)$ — максимума, то значение всего выражения будет минимальным. Указанные два условия могут выполняться одновременно, если одновременно выполняются два следующих утверждения:

- строки s_1 и s_3 не имеют общих символов;
- строки s_1 и s_3 входят в качестве подстрок в s_2.

Тогда:

$$r(s_1, s_3) = \max\{\|s_1\|, \|s_3\|\} = \|s_1\|,$$
$$r(s_1, s_2) = \|s_3\| + \|C\|, \; r(s_2, s_3) = \|s_1\| + \|C\|,$$

где C — вспомогательная строка, и, таким образом, минимальное значение выражения (2) можно записать в следующем виде:

$$\frac{\|s_3\| + \|C\| + \|s_1\| + \|C\|}{\|s_3\| + \|s_1\| + \|C\|} - \frac{\|s_1\|}{\|s_1\|} = \frac{\|C\|}{\|s_3\| + \|s_1\| + \|C\|} \geqslant 0.$$

Следовательно, в третьем случае для функции $r(s_1, s_3)$ также выполняется неравенство треугольника. Остальные случаи аналогичны уже рассмотренным. Таким образом, функция $r(s_1, s_2)$ является метрикой, заданной на множестве строк. Теорема доказана.

Замечание. Функция $r(s_1, s_2)$ принадлежит отрезку [0, 1] для любых строк s_1 и s_2.

В предложенном алгоритме данная метрика применяется для работы со строковыми реквизитами физических лиц, к которым относятся ФИО, адрес, документ и т. д. В связи с этим построенная с использованием данной метрики лингвистическая переменная позволяет обрабатывать запросы поиска для человека, похожего на другого человека, по реквизитам. Приняв от пользователя такой запрос, мы фактически получаем два значения: значение искомого реквизита и радиус поиска.

Алгоритм нечеткого поиска реквизитов физических лиц

Укрупненная блок-схема разработанного алгоритма нечеткого поиска реквизитов физических лиц в базах данных представлена на рис. 1.

В реализации алгоритма на языке PL-SQL СУБД Oracle 11g за предварительную выборку всех записей, отдаленно похожих на искомую, отвечает блок "Запрос количества идентичных людей в базе данных". Этот блок работает

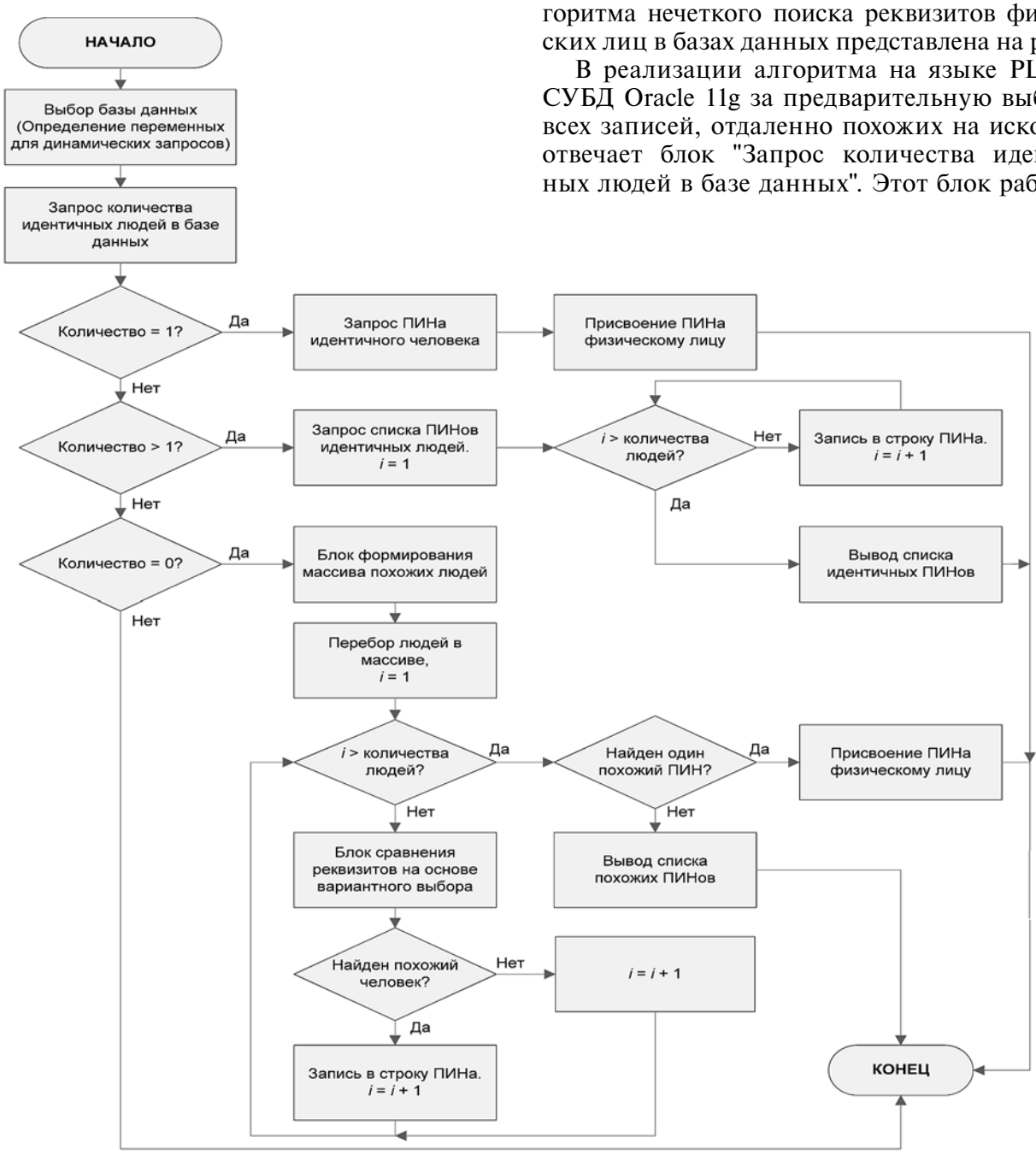

Рис. 1. Укрупненная блок-схема алгоритма нечеткого поиска реквизитов физических лиц в базах данных

по алгоритму прямого частичного сравнения разных наборов реквизитов, например, имени, отчества и даты рождения, формируя, тем самым, рабочий набор данных для рассматриваемого алгоритма идентификации. Затем в работу вступает "Блок сравнения реквизитов", ключевые функции которого отводятся логически выделенным процедурам COMPARISON_STRING и COMPARISON_NUMBER, созданным на основе модифицированного метода вычисления метрики Левенштейна, которые позволяют проводить интеллектуальное сравнение двух похожих строк или чисел, с учетом возможных неточностей или ошибок ввода. С помощью указанных процедур программа формирует набор совпадений и по результатам обработки предлагаемой и искомой записи выносит решение об идентичности строк. Например, у человека совпадает имя, отчество, дата рождения и номер паспорта, а в фамилии допущена ошибка в одну букву. В таком случае программа однозначно идентифицирует реквизиты. Данные процедуры могут применяться не только для идентификации реквизитов, но также везде, где требуется полнотекстовый поиск с нечетко заданными входными данными.

Алгоритм идентификации аккумулирует так называемый "опыт прошлых идентификаций" и записывает его в специально отведенное место в базе данных для использования в последующих идентификациях. Это позволяет сохранить не только результаты автоматической работы программы, но и решения операторов после отработки ими оставшихся не найденных реквизитов.

Технические и экономические показатели алгоритма

Для сравнительного анализа разработанного алгоритма рассмотрим метод на основе прямого сравнения. При использовании данной технологии упор идет на скорость обработки данных, а не на качество принятия решения системой. В итоге, после окончания работы процедуры на основе прямого сравнения остается много данных (около 20—30 % от общего числа строк), не связанных с исходными, которые необходимо отрабатывать вручную, что крайне затруднительно при больших объемах обрабатываемых данных.

При экспериментальном сравнении рабочих показателей двух алгоритмов получены следующие результаты.

Алгоритм прямого сравнения:

скорость обработки данных: ~ 100 000 строк в час;

точность идентификации (вероятность точного поиска реквизитов): ~ 80 %.

Алгоритм идентификации на основе нечеткого сравнения:

скорость обработки данных: ~ 80 000 строк в час;

точность идентификации (вероятность точного поиска реквизитов) ~ 99,9 %.

Отсюда можно сделать вывод, что у разработанного алгоритма минимизирована работа оператора по ручной отработке результатов, т. е. хотя скорость обработки несколько меньше, но алгоритм позволяет существенно разгрузить операторов за счет интеллектуальной системы принятия решений, чего не может предложить алгоритм прямого сравнения.

При сравнении экономических характеристик разработанного программного обеспечения на основе описываемого алгоритма с процедурой прямого сравнения для годового объема нечеткого поиска в 1 200 000 физических лиц были получены следующие данные: трудовые затраты на обработку информации по методу нечеткого сравнения по сравнению с методом прямого сравнения уменьшены в 6,7 раза; абсолютное снижение трудовых затрат составило 1446 ч; годовые затраты при использовании метода нечеткого сравнения уменьшились в 3 раза по сравнению с аналогичным периодом применения метода прямого сравнения, а годовой экономический эффект превысил 580 000 руб. Для наглядности некоторые стоимостные показатели, формирующиеся при использовании разработанного и применявшегося до настоящего времени программного обеспечения, отображены на диаграмме, приведенной на рис. 2 (см. вторую сторону обложки). Значения затрат отложены по оси ординат в рублях.

Заключение

Рассмотренный алгоритм нечеткого поиска персональных данных позволяет выполнять поиск реквизитов физических лиц в базах данных, используя данные ранее проведенного поиска, имеет высокую точность поиска и скорость работы по сравнению с методом прямого сравнения. Встроенная система приоритета реквизитов позволяет идентифицировать человека в таких случаях, как смена фамилии, имени, переезд, ошибки при ручном вводе данных, а также при частично отсутствующих реквизитах.

В перспективе данный алгоритм имеет возможность успешного внедрения в системы глобального объединения хранилищ государственных или коммерческих организаций, для ведения единой базы данных населения любой страны мира. Логическая структура разработанного алгоритма позволяет реализовать его на любом популярном языке программирования. Масштабируемость алгоритма позволяет применять программные процедуры на его основе, как в малых организациях, так и в крупных корпорациях, везде, где ведется и актуализируется реестр данных физических лиц. Возможные примеры использования: портал госуслуг; медицинские электронные системы; кадровые и бухгалтерские системы учета служащих; банковские системы хранения данных о клиентах и т. п.

Алгоритм реализован на языке PL-SQL системы управления базами данных Oracle 11g. Разработанное программное обеспечение, реализующее алгоритм нечеткого поиска персональных данных, внедрено и успешно функционирует с 2007 г. в нескольких муниципальных учреждениях г. Тольятти Самарской области.

Список литературы

1. **Международный** фонд автоматической идентификации. Технологии автоматической идентификации. URL: http://www.fond-ai.ru/art1/art223.html (дата обращения: 28.01.2018).

2. **Желудков А. В., Макаров Д. В., Фадеев П. В.** Особенности алгоритмов нечеткого поиска // Инженерный вестник МГТУ им. Н. Э. Баумана, 2014. С. 502—503.

3. **Soundex** метод нечеткого поиска. URL: https://ru.wikipedia.org/wiki/Soundex (дата обращения: 28.01.2018).

4. **Харитоненков А. В.** Поиск на неточное соответствие: коды Хемминга. URL: http://www.jurnal.org/articles/2009/inf32.html (дата обращения: 28.01.2018).

5. **Двоичный** алгоритм поиска подстроки. URL: https://ru.wikipedia.org/wiki/Двоичный_алгоритм_поиска_подстроки (дата обращения: 28.01.2018).

6. **Задача** о редакционном расстоянии, алгоритм Вагнера-Фишера. URL: http://neerc.ifmo.ru/wiki/index.php?title=Задача_о_редакционном_расстоянии,_алгоритм_Вагнера-Фишера (дата обращения: 28.01.2018).

7. **Расстояние** Дамерау — Левенштейна. URL: https://ru.wikipedia.org/wiki/Расстояние_Дамерау_—_Левенштейна (дата обращения: 28.01.2018).

8. **Левенштейн В. И.** Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады Академии наук СССР. 1965. Т. 163, № 4. С. 845—848.

9. **Бойцов Л. М.** Анализ строк. URL: http://itman.narod.ru/articles/infoscope/string_search.1-3.html (дата обращения: 28.01.2018).

N. I. Limanova, D. Sc., Professor, **M. N. Sedov,** Postgraduate Student, e-mail: SedovMN@inbox.ru,
Volga State University of Telecommunications and Informatics, Samara, Russian Federation

Fuzzy Searching Algorithm of Personal Details on the Basis of Levenshtein Distance

During the information exchange from one department to another there is a problem of personal identification. This problem concerns people who have partially or completely not coinciding personal details. In the represented work the new algorithm for identification of such people is elaborated. The algorithm is based on the fuzzy comparison and the metrics of Levenshtein. It allows us to find persons who have partial or complete not matching in surnames, names and other requisites in databases. The algorithm is implemented in PL-SQL in the Oracle database 11g.

Keywords: *interdepartmental exchange of information; fuzzy comparison; search of personal details; function of intellectual matching; personal identification number (PIN)*

DOI: 10.17587/it.24.523-528

References

1. **Mezhdunarodnyj** fond avtomaticheskoj identifikacii. Tehnologii avtomaticheskoj identifikacii (International Fund for Automatic Identification. Automatic identification technologies), available at: http://www.fond-ai.ru/art1/art223.html, free. rus. lang (date of access: 28.01.2018). (in Russian).

2. **Zheludkov A. V., Makarov D. V., Fadeev P. V.** Osobennosti algoritmov nechjotkogo poiska (Features of fuzzy search algorithms). Moscow, Inzhenernyj vestnik MGTU im. N. Je. Baumana, 2014, pp. 502—503 (in Russian).

3. **Soundex** metod nechjotkogo poiska (Soundex fuzzy search method), available at: https://ru.wikipedia.org/wiki/Soundex (date of access: 28.01.2018) (in Russian).

4. **Haritonenkov A. V.** Poisk na netochnoe sootvetstvie: kody Hemming (Search for inaccurate matching: Hamming codes), available at: http://www.jurnal.org/articles/2009/inf32.html (date of access: 28.01.2018) (in Russian).

5. **Dvoichnyj** algoritm poiska podstroki (Binary search algorithm for substring), available at: https://ru.wikipedia.org/wiki/Двоичный_алгоритм_поиска_подстроки (date of access: 28.01.2018) (in Russian).

6. **Zadacha** o redakcionnom rasstojanii, algoritm Vagnera-Fishera (The problem of the editorial distance, Wagner-Fisher algorithm), available at: http://neerc.ifmo.ru/wiki/index.php?title=Задача_о_редакционном_расстоянии,_алгоритм_Вагнера-Фишера (date of access: 28.01.2018) (in Russian).

7. **Rasstojanie** Damerau — Levenshtejna (Distance Dhamerau — Levenshtein), available at: https://ru.wikipedia.org/wiki/Расстояние_Дамерау_—_Левенштейна (date of access: 28.01.2018) (in Russian).

8. **Levenshtejn V. I.** Dvoichnye kody s ispravleniem vypadenij, vstavok i zameshhenij simvolov (Binary codes with correction of fallouts, inserts and substitutions of symbols), *Doklady Akademii nauk SSSR*, 1965, vol. 163, no. 4, pp. 845—848 (in Russian).

9. **Bojcov L. M.** Analiz strok (Analysis of strings), available at: http://itman.narod.ru/articles/infoscope/string_search.1-3.html. (date of access: 28.01.2018) (in Russian).