Метод, алгоритм и программное обеспечение для нечёткого поиска в базах данных

Н.И. Лиманова[1], М.Н. Седов[1]

[1]Поволжский государственный университет телекоммуникаций и информатики, Л.Толстого 23, Самара, Россия, 443010

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

1. Введение

В процессе межведомственного информационного обмена возникает проблема согласования основных реквизитов (ФИО, даты рождения, адреса, паспортных данных и т.п.) физических лиц в базах данных различных ведомств, обменивающихся информацией. Проблема персональной идентификации приобретает наибольшую актуальность для физических лиц, у которых частично или полностью не совпадают реквизиты.

Для оптимального управления большими массивами данных, связанных с реквизитами физических лиц, необходимо обеспечивать централизованные регламенты хранения таких характеристик, как ФИО, дата рождения, адрес, паспортные данные и т.д. В последнее время различные ведомства – держатели локальных баз данных стремятся объединить массивы для упрощения и повышения качества работы. Но возникает проблема сопоставления реквизитов физических лиц из одной базы данных реквизитам в другой. На помощь приходит интеллектуальный алгоритм поиска физических лиц в базах данных, или, по-другому, идентификация реквизитов физических лиц.

Для удобства обработки данных, каждому набору реквизитов в базах данных присваивается так называемый персональный идентификационный номер (ПИН). В случае обработки или передачи данных о физическом лице вся привязка осуществляется именно к этому ПИНу. В России, к сожалению, пока нет единой базы с реквизитами всех жителей, и, поэтому, в разных ведомствах ведется свой отдельный реестр физических лиц, и заводятся свои ПИНы. Проблема возникает при осуществлении обмена информацией о жителях между организациями, т.к. необходимо выполнить привязку входящих реквизитов к уже имеющимся. Для однозначной привязки необходимо выполнить интеллектуальный поиск физического лица в базе-приёмнике, который должен учитывать множество факторов: и потенциальные ошибки при ручном вводе, и отсутствующие или устаревшие реквизиты и т.п. Естественно предположить, что подобный поиск целесообразно реализовать в виде специализированного программного обеспечения.

2. Задача автоматизированного поиска

Традиционно данная проблема решается путём анализа тождественности основных реквизитов физического лица. Таких реквизитов несколько: фамилия, имя, отчество, дата рождения, серия, номер паспорта и адрес. Однозначно определив совпадение существующих и новых реквизитов, можно выполнить идентификацию физического лица в базе данных. Данный метод поиска выполняется вручную только в том случае, когда объём передаваемой информации невелик (количество физических лиц не более 30). При больших объёмах передаваемых данных используется автоматизированное сравнение тождественности реквизитов. Такой подход позволяет определить в среднем (50 – 60)% от общего числа идентифицируемых физических лиц. Оставшиеся (40 – 50)% представляют собой персональные данные, в которых частично или полностью не совпадают реквизиты. Такую информацию вручную обрабатывать ещё сложнее. Исходя из этого, задача автоматизированного поиска распадается на три подзадачи в зависимости от типа исходных данных. В результате сравнения могут получиться следующие три типа результатов.

1. Человек найден. Этот вывод может сформироваться в результате прямого сравнения реквизитов, а также равенства совокупностей некоторых ключевых данных. В данном случае физическое лицо сразу привязывается к соответствующему ПИНу.

2. Человек неоднозначно определён. Этот результат выводится при наличии ошибок, как в новых данных, так и в ранее полученных. Например, возможны ошибки оператора при ручном вводе основных реквизитов, порча данных при передаче, некорректная работа пакетных запросов при обработке информации и т.д. В данном случае выводится список ПИНов, основные реквизиты которых наиболее приближены к идентифицируемым данным.

3. Человек не найден. Этот случай свидетельствует о том, что данное физическое лицо отсутствует в базе и для привязки этого человека к ПИНу необходимо добавить его в имеющийся набор данных с присвоением ему нового ПИНа.

При создании автоматизированного комплекса программного обеспечения, который дает вышеперечисленные результаты, наиболее важным было достоверно определять границы между первым и вторым случаями, а также между вторым и третьим. Программное обеспечение, работающее без подобного разграничения, всем найденным лицам однозначно проставит ПИНы, а те, чьи данные определены неоднозначно, выводятся в отчёт для ручной обработки оператором. При этом все не найденные лица добавятся в базу с присвоением нового ПИНа. На исправление последствий работы такого программного обеспечения уйдёт немало времени и сил у большого количества компетентных служащих учреждения.

Неправильная идентификация может привести также к большому количеству данных в отчёте для ручной отработки, к присвоению ПИНа не тому человеку и к добавлению излишних данных. Последствия таких ошибок в худшем случае могут полностью парализовать работу учреждения на неопределённое время, в лучшем – отнять более 10% рабочего времени специалистов на исправление ошибок. Анализ существующего программного обеспечения показал, что единого идентификатора нет, универсальный алгоритм идентификации также отсутствует.

3. Математическая модель

Известно несколько видов метрик, отражающих интуитивное понятие схожести строк. Наиболее распространены расстояния Хемминга, метрика Левенштейна и расстояние редактирования.

Расстояние Хемминга определяется для строк одинаковой длины и задаётся как число позиций, в которой символы не совпадают. Фактически, расстояние Хемминга рассчитывается как минимальная цена преобразования одной строки в другую, когда возможна только одна операция редактирования строк – замена.

В случае, когда требуется произвести сравнение строк разной длины, используются метрика Левенштейна или расстояние редактирования. Эти две метрики очень похожи по построению и фактически являются одной и той же метрикой, несколько модифицированной для каждого случая. Так, например, метрика Левенштейна определяется как минимальная цена

преобразования одной строки в другую с использованием трёх операций: вставки, замены и удаления символа, причём все три операции имеют одинаковый вес.

Расстояние редактирования является модификацией метрики Левенштейна на случай, когда разрешены всего две операции: вставки и удаления.

В связи с вышеизложенным, была выбрана именно общая метрика Левенштейна, которая поддерживает все три операции со строкой. 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 при помощи величины $p(s_1,s_2)$ метрику Левенштейна, а величиной $||s_i||$ – длину строки s_i. Тогда функция:

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

является метрикой.

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

$p(s_1,s_2) \geq 0$,

$p(s_1,s_2) = p(s_1,s_2)$,

$p(s_1,s_2) + p(s_2,s_3) \geq p(s_1,s_3)$

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:

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

Запишем это неравенство в виде:

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

Возможны следующие случаи:

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

Рассмотрим первый случай. Имеем:

$$\frac{p(s_1, s_2)}{\max\{|| s_1 ||, || s_2 ||\}} + \frac{p(s_2, s_3)}{\max\{|| s_2 ||, || s_3 ||\}} - \frac{p(s_1, s_3)}{\max\{|| s_1 ||, || s_3 ||\}} = \frac{p(s_1, s_2)}{|| s_2 ||} + \frac{p(s_2, s_3)}{|| s_3 ||} - \frac{p(s_1, s_3)}{|| s_3 ||} \geq$$

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

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:

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

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:

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

Тогда:

$r(s_1,s_3) = \max\{||s_1||,||s_3||\}=||s_1||$,

$r(s_1,s_2) = ||s_3|| + ||C||$, $r(s_2,s_3) = ||s_1|| + ||C||$,

и, таким образом, минимальное значение выражения (2) запишется в виде:

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

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. Поэтому 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.

4. Алгоритм поиска физических лиц в базах данных на основе нечёткого сравнения

Укрупненная блок-схема разработанного алгоритма поиска физических лиц в базах данных приведена на рисунке 1. Предлагаемый алгоритм, представленный в виде процесса Data Mining, включает следующие этапы:

1. анализ предметной области;
2. постановка задачи;
3. подготовка данных;
4. построение моделей;
5. проверка и оценка моделей;
6. выбор модели;
7. применение модели;
8. коррекция и обновление модели.

Рассмотрим указанные этапы более детально.

1. Предметная область представляет собой наборы данных с основными реквизитами физических лиц в разных организациях и ведомствах.

2. Задача поиска заключается в том, чтобы в условиях отсутствия единого персонального идентификационного номера производить поиск набора реквизитов в одной базе данных в соответствии с реквизитами физического лица в другой базе данных.

3. Подготовка данных представляет собой организацию укрупнённой выборки, включающей около 300-500 наборов, отдалённо похожих на искомый. Ниже приведен фрагмент кода, позволяющий организовать программно такую выборку:

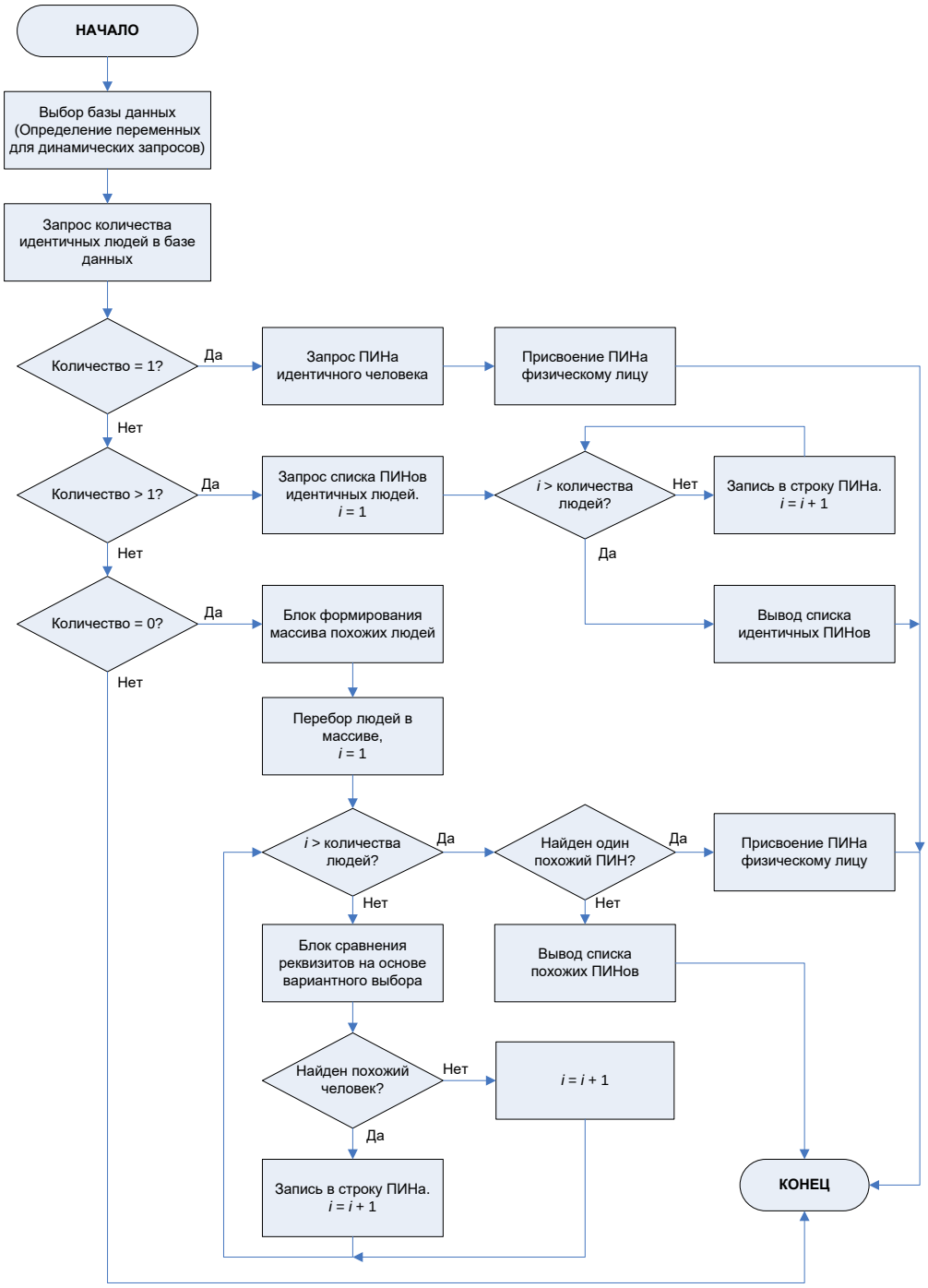

Рисунок 1. Укрупненная блок-схема разработанного алгоритма поиска физических лиц в базах данных.

```
    CURSOR persons IS
  SELECT p.person_id, p.lastname, p.firstname, p.patronymic, p.birthdate
    FROM work.person p
WHERE (((SOUNDEX(TO_TRANSLIT(p.lastname))= SOUNDEX(TO_TRANSLIT(fo_Lastname)))
  AND (SOUNDEX(TO_TRANSLIT(p.firstname)) =  SOUNDEX(TO_TRANSLIT(fo_Firstname))))
   OR  ((SOUNDEX(TO_TRANSLIT(p.lastname))  =   SOUNDEX(TO_TRANSLIT(fo_Lastname)))
AND (SOUNDEX(TO_TRANSLIT(p.patronymic))= SOUNDEX(TO_TRANSLIT(fo_Patronymic))))
 OR ((SOUNDEX(TO_TRANSLIT(p.firstname)) = SOUNDEX(TO_TRANSLIT(fo_Firstname)))
```

AND(SOUNDEX(TO_TRANSLIT(p.patronymic))=SOUNDEX(TO_TRANSLIT(fo_Patronymic)))));

4. Построение моделей заключается в выявлении закономерностей при анализе данных, полученных в результате этапа 3, проявившихся именно в этом наборе данных и, возможно, подходящих для будущих наборов.

5. Проверка и оценка моделей представляет собой тестирование закономерностей на количество удовлетворяющих им наборов данных. Чем больше наборов подходят для конкретной модели, тем более ценной становится выявленная закономерность.

6. Выбор модели заключается в выявлении наиболее значимых закономерностей для дальнейшего использования при будущих запусках процедуры идентификации.

7. Применение модели представляет собой использование закономерности, полученной и утверждённой при прошлом запуске процедуры идентификации, на текущих наборах данных.

8. Коррекция и обновление модели заключается в анализе результата приложения закономерности к новому набору данных, и, при необходимости, коррекция модели для расширения круга подходящих наборов при нечётком поиске соответствия реквизитов физического лица.

Программно это выглядит примерно так (с использованием динамического SQL):

```
 -- Выполняем быструю идентификацию
OPEN cur_Ref_fast_ident FOR
'SELECT t.'||v_Col_pin||'   FROM '||v_Table||' t
WHERE UPPER(TRIM(t.'||v_Col_lastname||')) =UPPER(TRIM('''||fo_Lastname||'''))
    AND UPPER(TRIM(t.'||v_Col_firstname||')) =UPPER(TRIM('''||fo_Firstname||'''))
AND NVL(UPPER(TRIM(t.'||v_Col_patronymic||')), "_") =NVL(UPPER(TRIM('''||fo_Patronymic||''')),
"_") AND t.'||v_Col_birthdate||' = '''||TO_CHAR(fo_Birthdate, 'dd.mm.yyyy')||'''';
FETCH cur_Ref_fast_ident  BULK COLLECT INTO c_fast_ident;  CLOSE cur_Ref_fast_ident;
 -- В зависимости от количества пинов идентичных людей
 IF (NVL(c_fast_ident.count, 0) = 1) THEN fout_Pin := c_fast_ident(1);
 ELSIF (NVL(c_fast_ident.count, 0) > 1) THEN FOR i IN c_fast_ident.first..c_fast_ident.last LOOP
            fout_Pin_list := fout_Pin_list||TO_CHAR(c_fast_ident(i))||' ';   END LOOP;
 -- Если быстрая идентификация не дала результатов
 ELSIF (NVL(c_fast_ident.count, 0) = 0) THEN
        -- Записываем данные из курсора в коллекцию
        OPEN cur_Ref_full_ident FOR v_Cur_ident;
          FETCH cur_Ref_full_ident BULK COLLECT INTO c_full_ident;
        CLOSE cur_Ref_full_ident;
        IF (NVL(c_full_ident.count, 0) > 0) THEN FOR i IN c_full_ident.first..c_full_ident.last
LOOP
   -- Выполняем полную идентификацию
   CASE   [Блок сравнения реквизитов на основе вариантного выбора (см. рисунок 1)]
    ELSE NULL;   END CASE;
```

В разработанной реализации алгоритма на языке PL-SQL СУБД Oracle 11g ключевые функции отводятся логически выделенным процедурам COMPARISON_STRING и COMPARISON_NUMBER, созданным на основе модифицированного метода вычисление метрики Левенштейна, которые позволяют проводить интеллектуальное сравнение двух похожих строк или чисел, с учётом возможных неточностей или ошибок ввода. Данные процедуры могут применяться не только для идентификации реквизитов, но также везде, где требуется полнотекстовый поиск с нечётко заданными входными данными.

5. Технические и экономические показатели алгоритма

Для сравнительного анализа разработанного алгоритма рассмотрим технологию идентификации на основе прямого сравнения. При использовании данной технологии упор идёт на скорость обработки записей, а не на качество принятия решения системой. В итоге, после окончания работы процедуры на основе прямого сравнения, остаётся много данных (около 20-30% от общего количества строк), не связанных с исходными, которые необходимо

отрабатывать вручную, что крайне затруднительно при больших объемах обрабатываемых данных.

При сравнении рабочих показателей двух алгоритмов выявлено:

Алгоритм прямого сравнения:

Скорость обработки данных: ~100000 строк в час;

Точность идентификации (вероятность точного поиска реквизитов): ~ 80%

Алгоритм идентификации на основе нечёткого сравнения:

Скорость обработки данных: ~80000 строк в час;

Точность идентификации (вероятность точного поиска реквизитов): ~ 99,9%

Отсюда можно сделать вывод, что у разработанного алгоритма минимизирована работа оператора по ручной отработки результатов, т.е. хотя скорость обработки несколько меньше, но алгоритм позволяет существенно разгрузить операторов за счёт 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 объёма идентификации в 1 200 000 физических лиц были получены следующие данные: трудовые затраты на обработку информации по методу нечёткого сравнения по сравнению с методом прямого сравнения уменьшены в 6,7 раза, абсолютное снижение трудовых затрат составило 1446 часов, годовые затраты при использовании метода нечёткого сравнения уменьшились в 3 раза по сравнению с аналогичным периодом применения метода прямого сравнения, а годовой экономический эффект превысил 580000 руб. Для наглядности некоторые стоимостные показатели, формирующиеся при использовании разработанного и применявшегося до настоящего времени программного обеспечения отображены на диаграмме, приведенной на рисунке 2. Величины затрат отложены по оси ординат в рублях.

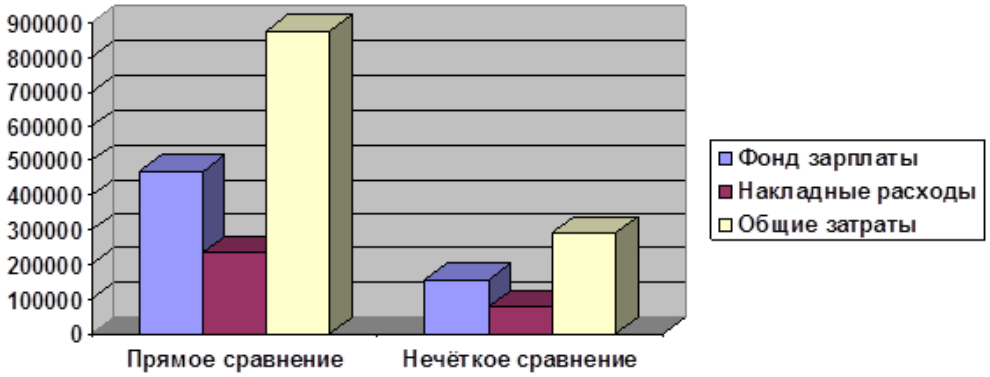

Рисунок 2. Диаграмма для сравнительного анализа стоимостных показателей при использовании методов прямого и нечёткого сравнения.

6. Заключение

Рассмотренный метод автоматизированного поиска персональных данных на основе нечёткого сравнения, спроектированный с использованием технологии Data Mining, позволяет быстро определять людей, используя данные ранее проведенного поиска. 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.

Самообучающиеся системы позволяют освободить человеческие ресурсы для выполнения творческих задач. В этой области технология Data Mining предоставляет полный набор теоретических и практических средств для выбора, разработки или использования интеллектуальных компьютерных систем.

Рассмотренную в статье процедуру идентификации можно рассматривать как часть системы поддержки принятия решений (СППР). Процедура не требует вмешательство оператора, накапливает опыт и самообучается в процессе работы, позволяя, тем самым, полностью

освободить специалистов от низкопрофильной, неэффективной, ручной работы напрямую с наборами реквизитов физических лиц, хранящимися в базах данных.

В перспективе, данный алгоритм обладает возможностью 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 ведётся и актуализируется реестр данных физических лиц. Возможные примеры использования: портал госуслуг, медицинские электронные системы, кадровые и бухгалтерские системы учёта 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 метод автоматизированного поиска персональных данных на основе нечёткого сравнения, внедрено и успешно функционирует с 2007 года в муниципальном учреждении «Городской информационный центр» г. Тольятти Самарской области.

7. Литература

[1] Международный фонд автоматической идентификации. Технологии автоматической идентификации [Электронный ресурс]. – Режим доступа: http://www.fond-ai.ru/art1/art223.html, свободный. яз. рус. (дата обращения 15.11.2017).

[2] Желудков, А.В. Особенности алгоритмов нечёткого поиска / А.В. Желудков, Д.В. Макаров, П.В. Фадеев // Москва, Инженерный вестник МГТУ им. Н.Э. Баумана, 2014. – С. 502-503.

[3] Soundex метод нечёткого поиска [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Soundex (дата обращения 15.11.2017).

[4] Харитоненков, А.В. Поиск на неточное соответствие: коды Хемминга / А.В. Харитоненков. – [Электронный ресурс]. – Режим доступа: http://www.jurnal.org/articles/2009/inf32.html (дата обращения 15.11.2017).

[5] Двоичный алгоритм поиска подстроки [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Двоичный_алгоритм_поиска_подстроки (дата обращения 15.11.2017).

[6] Задача о редакционном расстоянии, алгоритм Вагнера-Фишера [Электронный ресурс]. – Режим доступа: http://neerc.ifmo.ru/wiki/index.php?title=Задача_о_редакционном_расстоянии,_алгоритм_Вагнера-Фишера (дата обращения 15.11.2017).

[7] Расстояние Дамерау-Левенштейна [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Расстояние_Дамерау_Левенштейна (дата обращения 15.11. 2017).

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

[9] Бойцов, Л.М. Анализ строк / Л.М. Бойцов. – [Электронный ресурс]. – Режим доступа: http://itman.narod.ru/articles/infoscope/string_search.1-3.html, свободный. яз. рус. (дата обращения 15.11.2017).

Method, algorithm and software for fuzzy search in databases

N.I. Limanova[1], M.N. Sedov[1]

[1]Povolzhskiy State University of Telecommunications and Informatics, Lev Tolstoy street 23, Samara, Russia, 443010

Abstract. During the information exchange from one department to another there is a problem of personal identification. This problem concerns the 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: indistinct matching, search of personal details, function of intellectual matching.