АЛГОРИТМ СОПОСТАВЛЕНИЯ ДЛЯ НОРМАЛИЗАЦИИ ДАННЫХ СИСТЕМЫ ФОРМИРОВАНИЯ ОПТИМАЛЬНЫХ ПРЕДЛОЖЕНИЙ НА ВЫБОР ИЗДЕЛИЙ ЭЛЕКТРОННОЙ КОМПОНЕНТНОЙ БАЗЫ В ПРОЦЕССАХ РАЗРАБОТКИ РАДИОЭЛЕКТРОННОЙ АППАРАТУРЫ

Ю.В. Рубцов (АО «ЦКБ «Дейтон»)

Рассмотрен процесс разработки алгоритма сопоставления для нормализации данных системы формирования оптимальных предложений на выбор изделий электронной компонентной базы при разработке радиоэлектронной аппаратуры. Рассмотрены оригинальные методы сопоставления данных и методы расчета показателей точности сопоставления. Алгоритм применяется в составе системе формирования предложений на выбор электронной компонентной базы в процессах разработки радиоэлектронной аппаратуры.

Ключевые слова: электронная компонентная база, радиоэлектронная аппаратура, интеллектуальный анализ данных, машинное обучение, алгоритм сопоставления данных, точность сопоставления.

ВВЕДЕНИЕ

Разрабатывая радиоэлектронную аппаратуру (РЭА), специалисты используют информацию об изделиях электронной компонентной базы (ЭКБ), которая для этого подготавливается: собирается, обрабатывается и анализируется. Одним из этапов обработки собранной информации является ее нормализация, в ходе которой выполняется структурирование информации в базе данных (БД) путем устранения избыточности и обеспечения целостности [1]. Нормализация упорядочивает информацию в БД перед использованием, оптимизирует ресурсы на обработку и хранение данных, снижает вероятность возникновения ошибок при их обработке [2]. Нормализация гарантирует, что каждый элемент данных хранится только в одном месте. Это обеспечивает качество данных в системе формирования предложений на выбор ЭКБ в процессах разработки РЭА (далее - Система).

Источниками информации об ЭКБ являются субъекты и объекты. Они доступны для идентификации (имеют наименование, контактные данные или точки доступа), имеют обязательства по предоставлению информации, информируют о происхождение информации и дают разрешение на ее использование в Системе.

Информация поступает в Систему от разработчиков и потребителей изделий ЭКБ, исследовательских, измерительных и испытательных организаций, а также из конструкторской и технологической документации на изделия, результатов исследований, измерений, испытаний, применения, эксплуатации, модернизации и утилизации. Перед началом нормализации данных важно убедиться, что информация, полученная от различных источников, является актуальной, достоверной и готовой к сопоставлению.

Целью сопоставления данных является выявление избыточности (устранение дубликатов), обеспечение актуальности данных, снижение вероятности возникновения аномалий данных, которые могут нарушить их целостность. Данные могут не являться идентичными, но передавать одну и ту же информацию. После сопоставления таких данных между ними устанавливаются связи.

Сопоставление данных выполняется на основе определенных критериев (идентифицирующих атрибутов) [3]. Критерии зависят от контекста и характера сопоставляемых данных [4].

Сопоставление данных включает: сравнение данных для установления их идентичности; анализ данных для установления их отличительных черт; выделение существенных свойств данных, общих для целей нормализации.

Сопоставление данных об ЭКБ вручную – длительная и низкоэффективная работа, поэтому разработка средств автоматизации для этой задачи является актуальной.

Сегодня, когда технологии для сопоставления данных находятся на подъеме, разработка систем управления информацией позволяет достичь целей, выходящих за рамки управления БД [5].

ИССЛЕДОВАНИЯ АЛГОРИТМОВ СОПОСТАВЛЕНИЯ ДАННЫХ

При проведении описываемой работы исследованы доступные алгоритмы сопоставления данных. Обобщены и проанализированы выполняемые ими функции, методы реализации. Исследовались алгоритмы сопоставления данных, обеспечивающие разумный баланс: установление слишком строгого правила соответствия дает пропуски действительных совпадений; свободные правила логически объединяют несвязанные данные.

В исследованиях учитывались:

- реальная ситуация роста наборов данных и требования алгоритмов к ресурсам. Сравнение миллионов записей в БД требуют высокой вычислительной мощности и оптимизированных алгоритмов;
- необходимость выявления и исправления ошибок в параметрах и показателях ЭКБ;
- сопоставление частично структурированного текста требует передовых методов, таких как разрешение сущностей на основе искусственного интеллекта (ИИ);
- наличие в источниках информации собственной структуры данных и правил их формирования.

В результате исследований сформирована схема разработки и проверки функционирования алгоритма сопоставления данных от ЭКБ для применения в Системе. Процесс разработки алгоритма для сопоставления данных об ЭКБ включает следующие этапы (рис. 1).

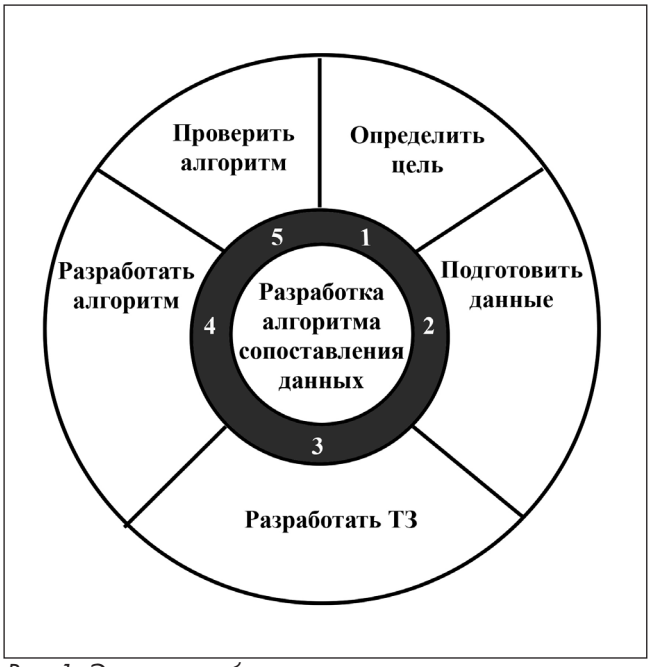

Рис. 1. Этапы разработки алгоритма для сопоставления данных об ЭКБ

1) Определение области действия алгоритма сопоставления данных. На этом этапе определяется цель: нормализация данных систем формирования оптимальных предложений на выбор изделий ЭКБ в процессах разработки РЭА, путем дедупликации данных, определения связей между данными и объединения данных собранных из разных источников.

2) Подготовка данных. Выполнение очистки данных, стандартизацию условных обозначений ЭКБ, кодов и классификационных признаков; удаление лишних символов из полей данных; единообразие форматов данных. Обеспечение единообразия и согласованности данных повышает качество и предотвращает ложноположительные и отрицательные результаты.

Очистка данных включает: устранение неоднозначностей, чтобы можно было получить стандартизированное и единообразное представление по всем источникам данных. На этом этапе выполняются:

а) удаление и замена пустых значений, начальных/конечных пробелов, избыточных символов и цифр, знаков препинания;

б) агрегирование данных, замена более длинных строк на более мелкие подкомпоненты;

в) трансформация регистров букв (верхние в нижние или из нижних в верхние) для обеспечения согласованного стандартизированного представления;

г) объединение одинаковых или похожих данных, чтобы избежать дублирования;

д) сопоставление и преобразование шаблонов значений данных для обеспечения согласованности.

3) Разработка технического задания на алгоритм сопоставления: перечень выполняемых функций, требования к параметрам, методам верификации (доказательства правильности функционирования).

4) Разработка алгоритма сопоставления: методы для перехода от имеющихся входных данных к требуемым выходным.

5) Верификация алгоритма – подтверждение соответствия требуемым параметрам.

Специалистами АО «ЦКБ «Дейтон» разработан алгоритм сопоставления данных. Используется структура данных в отношении определенного класса ЭКБ – корпусов – компоненты, защищающие кристаллы, обеспечивающие связь с внешними цепями (свидетельство о регистрации БД[1]). Программные средства реализации алгоритма представлены свидетельствами на регистрацию программ для ЭВМ[2].

[1] Свидетельство о государственной регистрации БД №2015621293 НА, Корпуса ЭКБ: заявитель АО «ЦКБ «Дейтон».

[2] Свидетельства о государственной регистрации программы для ЭВМ: №2023614155, Программа визуализации данных об изделиях электронной техники; №2022683437, Программа формирования базы данных об изделиях электронной техники; №2022668891, Программа управления данными об изделиях электронной техники. Заявитель АО «ЦКБ «Дейтон».

МЕТОДЫ СОПОСТАВЛЕНИЯ ДАННЫХ

Рассмотрим методы сопоставления данных, применяемые в разработанном алгоритме: точное сопоставление, неточное сопоставление или сложные сопоставления, учитывающие существенные различия в данных. При работе со сложными данными и для обеспечения высокой точности используется подход, основанный на технологиях ИИ и машинного обучения (МО).

Методы точного сопоставления данных

Данные идентичные, если их значения точно совпадают. Сопоставление идентичных данных (СИД) эффективно используется, когда элементы данных согласованы и структурированы. Методы СИД позволяют связывать точные атрибуты в наборах данных. При выполнении алгоритма СИД незначительные изменения в формате данных могут привести к несоответствию. Эти методы включают сравнение полей данных, которые должны быть идентичными для сопоставления. Точное сопоставление идеально подходит для работы с четко определенными атрибутами, но оно может оказаться недостаточным при наличии вариаций, опечаток или неточных данных.

Чаще всего самый весомый вклад в результаты точного сопоставления данных вносит синтаксический подход (совпадение наименований ЭКБ, кодов классификации). Методы синтаксического сравнения строк определяют сходство между элементами, основываясь только на эквивалентности названий. Наиболее простой метод для определения равенства названий – точное совпадение строк. Примером алгоритмов СИД являются алгоритмы двоичного поиска. Другой, более универсальный вариант - это использование методов неточного сопоставления данных.

Методы неточного сопоставления данных

Помимо синтаксической информации, содержащейся в данных об ЭКБ, важную роль играет семантическая информация: формат и типы данных, допустимые значения, ссылочные ограничения и т. д.

Данные неидентичные, если их значения совпадают неточно. Сопоставление неидентичных данных (СНИД) учитывает вероятность наличия общих черт на основе заранее определенных критериев или правил.

Семантическое соответствие обусловлено также терминологическими отношениями (синонимы, гиперонимы и гипонимы). Например, слова «резистор» и «сопротивление» в отношении изделий ЭКБ являются синонимами и, следовательно, соответствуют друг другу. Так же, как и их возможный потенциал: 5 кОм и 5 kΩ. Это требует использования дополнительных источников, таких как словари, тезаурусы, онтологии.

СНИД используется при работе с противоречивыми или неполными данными. Неточное сопоставление позволяет сравнивать данные на основе предопределенных правил или критериев. Примерами алгоритмов неточного сопоставления являются алгоритмы расстояния Левенштейна и расстояния Яро-Ринклера для количественной оценки сходства между строками. Они позволяют находить совпадения с расхождениями. Методы СНИД предназначены для обработки частичных совпадений, обеспечивая гибкость при работе с реальными искажениями данных. Предоставляя оценку сходства, такую как веса, а не строгое совпадение «да/нет», СНИД позволяет принимать более правильные решения и повышать устойчивость к несовершенствам собранных данных.

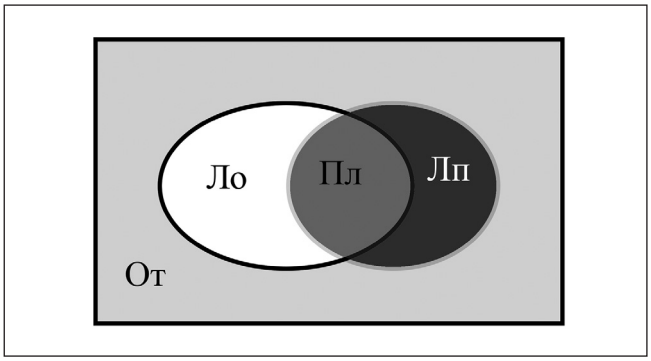

Рис. 2. Варианты сопоставления данных

Сложные сопоставление, учитывающие существенные различия в данных

Для сложных сопоставлений, учитывающих существенные различия в данных, используются технологии машинного обучения, позволяющие обучать Систему на основе закономерностей в данных и адаптировать к нюансированным изменениям. Для изучения закономерностей и вывода совпадений на основе неструктурированных или сложных данных используются нейронные сети.

В процессе сопоставления данным присваиваются значения или веса на основе их релевантности или вероятностей. Затем вычисляется показатель сходства между совпадающими записями данных с помощью методов косинусного сходства, евклидова расстояния, индекса Жаккара или расстояния Хемминга. Различные метрики сходства используются для количественной оценки того, насколько данные совпадают. Рассчитывается общий вес, полученный после сопоставления различных данных об ЭКБ. Определяется пороговая оценка, которая представляет точность процесса сопоставления данных.

МЕТОДЫ РАСЧЕТА ПОКАЗАТЕЛЕЙ ТОЧНОСТИ СОПОСТАВЛЕНИЯ

Далее, прежде чем использовать обработанные с помощью рассмотренных методов данные, их необходимо оценить по ряду выделенных параметров.

1) Точность – это близость вариантов, сопоставляемых данных к истинному результату. Точность отражает долю данных с правильными сопоставлениями среди всех найденных. Для определения точности функционирования алгоритма сопоставления данных введём двумерные пространства вариантов сопоставления данных: Пл – истинных положительных (правильных) сопоставлений, Лп – ложных положительных сопоставлений, Ло – ложных отрицательных и От – истинных отрицательных. Визуально они представлены на рис. 2. Математическое выражение для расчета параметра точности:

$$T = \frac{Пл}{Пл+Лп+Ло+От}. \quad (1)$$

Этот показатель имеет ряд недостатков: а) он не учитывает дисбаланс вариантов; б) не даёт информацию о типе ошибок; в) зависит от порога вариации, изменение которого может значительно повлиять на точность; г) не учитывает особенностей сопоставления изменяемых данных при развитии (модернизации, доработке) ЭКБ.

Для устранения перечисленных недостатков вводится мера для оценки точности как взвешенное среднее:

$$T(\alpha) = \frac{Пл}{Пл+(1-\alpha)*Ло+\alpha*Лп}. \quad (2)$$

где α - коэффициент, позволяющий задавать относительную значимость точности, изменяется в диапазоне 0...1.

Метод применения меры для оценки точности показывает, насколько точно работает алгоритм для различных наборов данных при сопоставлении, позволяет усреднить показатель и оценить алгоритм, обеспечивающий сопоставление данных для нормализации и оптимального выбора изделий ЭКБ для применения в РЭА. Основное отличие заключается в том, что для расчёта не используется линейная интерполяция. Вместо этого формируется средневзвешенное значение, а в качестве веса используется относительная значимость точности.

2) Эффективность – определяется объёмом ресурсов, необходимых для выполнения задач сопоставления данных. Эффективность алгоритма оценивается по двум критериям:

а) время выполнения. Зависит от числа выполняемых алгоритмом операций, вычислительных ресурсов, объёмов сопоставляемых данных;

б) объём используемой памяти. Зависит от числа переменных, типа и размера структуры данных, числа вызываемых функций и способа выделения памяти, объёмов сопоставляемых данных.

Сопоставление данных требует больших вычислительных ресурсов. При наличии миллионов записей в наборе данных сравнение внутри и между наборами данных, а затем сопоставление по нескольким полям может создать нагрузку на Систему и занять много времени для получения результата. По этой причине применяются методы блокировки или индексирования, чтобы убрать некоторые записи из процесса сравнения. Сравнения блокируются путем выбора атрибута, который с высокой вероятностью будет одинаковым для двух записей, если они принадлежат одному и тому же изделию. Если значения двух наборов данных слишком сильно различаются, записи не рассматриваются в процессах сопоставления.

Для сопоставления необходимо сравнить данные, описывающие одну и ту же информацию. Сравнение нужно, так как разные источники могут дать разную информацию для:

а) структуры данных, например, один источник хранит функциональное назначение изделия в одном поле, а другой – в полях: функциональное назначение и область применения;

б) содержания. Для этих целей выполняется парсинг данных (на начальном этапе подготовки данных) для разделения более длинного поля на составляющие, например, величина параметра и единица измерения.

Для решения задач сопоставления данных алгоритму необходима память для: хранения программного кода алгоритма; входных и выходных данных; вычислительного процесса включая: переменные и стековое пространство для вызова подпрограмм, которое может быть существенным при использовании рекурсии и в целях использования результата одного шага для подсчитывания другого. Такой подход делает программный код проще, короче и быстрее в выполнении, чем альтернативные варианты.

Для достижения максимальной эффективности необходимо уменьшить использование ресурсов:

$$Э = \frac{1}{Ор} * 100\%, \quad (3)$$

где *Ор* – объём требуемых ресурсов для функционирования алгоритма.

Для оценки эффективности алгоритма сравнения данных применяется бенчмаркинг - сравнение показателей при тестировании алгоритма. Сравнивается время выполнения и требуемые ресурсы памяти тестируемого алгоритма на разных образцах входных данных. Тестировать поток исполнения можно как простыми методами (расставлять везде операторы печати и проверять вывод), так и сложными (составлять специальные тестовые примеры поведения алгоритма).

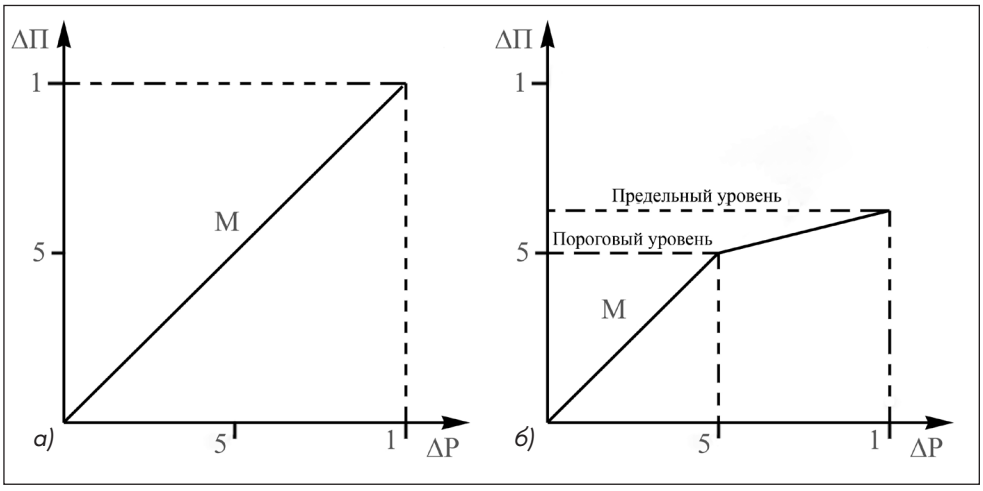

Рис. 3. Примеры решения задачи масштабирования: а) идеальный вариант; б) с незначительным повышением производительности алгоритма

Требуемые ресурсы необходимо привести к условным единым единицам измерения, или определить какой параметр наиболее важен, например, требование высокой скорости. Для этого необходимо оценить время (или число шагов), необходимые для решения задачи размера *n*. Искомый параметр представляется выражением:

$$T(n) = 4n^2 - 2n + 2. \quad (4)$$

По мере роста *n* его квадрат становится доминирующим, и всеми остальными составляющими можно пренебречь. Такой подход основан на методе «Big O». Определяется основная операция, которая вносит наибольший вклад в выполняемую задачу. В нашем случае – задача сложного сопоставления данных – наиболее трудоемкая операция во внутреннем цикле алгоритма.

$$T(n) \in O(n^2). \quad (5)$$

Отметим, что время выполнения алгоритма может зависеть от формата входных данных. Например, алгоритм сопоставления работает быстрее на отсортированной последовательности входных данных. Для повышения требуемой эффективности необходимо тестировать алгоритм при увеличении объемов входных данных, оптимизировать программный код и разрабатывать оптимальные структуры входных данных.

3) Масштабируемость - способность решать задачи сопоставления данных с увеличением рабочей нагрузки при добавлении ресурсов (создании параллельных процессов). Получена математическая зависимость масштабируемости:

$$M = \frac{\Delta П}{\Delta Р}, \quad (6)$$

где $\Delta П$ – прирост рабочей нагрузки; $\Delta Р$ – добавление ресурсов.

Чем ближе отношение (6) к единице, тем эффективнее решается задача масштабируемости в алгоритме. На рис. 3а показан пример идеального варианта решения задачи масштабируемости. Прирост рабочей нагрузки и требуемых ресурсов изменяются пропорционально в интервале 0...1.

Масштабируемость показывает способность алгоритма наращивать дополнительные ресурсы без структурных изменений в Системе. В алгоритмах с плохой масштабируемостью добавление ресурсов приводит лишь к незначительному повышению производительности (рис. 3б). С определенного «порогового» уровня добавление ресурсов снижает полезный эффект.

Масштабируемость показывает способность алгоритма выполнять сопоставление больших объемов данных об ЭКБ, учитывая их потенциальный рост.

4) Гибкость – предлагает настраиваемые правила сопоставления, когда алгоритм работает с различными форматами и структурами данных. Для выполнения сопоставления данных об изделиях ЭКБ гибкость алгоритма требует использовать:

а) одномерный массив – строка с данными, например, перечень корпусов, применяемых для одной микросхемы;

б) многомерный массив – комбинация из нескольких одномерных массивов, например, коэффициент усиления для микросхемы операционного усилителя при различной температуре среды эксплуатации;

в) динамический массив – при его создании задаётся максимальная величина и число заполненных элементов. При добавлении новых элементов они сначала заполняют массив до максимальной величины, а при превышении создаётся новый массив, с большей максимальной величиной, например, параметры надежности ЭКБ по результатам различных условий эксплуатации;

г) связанный список – группа из узлов. В каждом узле содержаться: данные, указатель на следующий узел, указатель на предыдущий узел. Алгоритм позволяет быстро перемещаться между элементами списка с помощью указателей, например, нетлист (список цепи) схемы включения изделия ЭКБ;

д) стек – позволяет добавлять и удалять элементы по принципу «последним пришёл - первым ушёл». Используется в Системе для временного хранения сопоставляемых данных;

е) очередь – в отличие от стека работает добавлением данных в конец и извлечением первых. Применяется в алгоритме для распараллеливания процессов, как показано в оценке масштабируемости алгоритма;

ж) множество – когда данные не упорядочены. Они хранятся в наборе. Их нельзя структурировать. Но с ними можно работать как с математическими множествами: объединять, искать пересечения, вычислять разность и оценивать, является ли одно множество подмножеством другого. Например, классификаторы справочных данных: единиц измерения, показателей качества, надежности, стран – изготовителей аналогов изделий;

з) карта поиска – ассоциативный массив. Данные в карте хранятся в паре ключ - значение, причем ключ уникален, значения могут повторяться. Например, значение выходного напряжения микросхемы цифроаналогового преобразователя при различных температурах окружающей среды;

и) дерево поиска – данные лежат в узлах. У каждого узла могут быть один или несколько дочерних и только один родитель, то есть они расходятся, как ветви дерева. Используется для хранения данных в отсортированном виде с возможностью быстрого их поиска. Например, классификатор изделий ЭКБ, представленный в соответствующих исследованиях [6];

к) префиксное дерево – данные хранятся последовательно: каждый узел – это префикс, по которому выполняется поиск следующих узлов. Например, условные обозначения изделий ЭКБ, которые можно разделить на префиксы. Например, для микросхем: категория качества, технологический уровень, функциональное назначение, применяемый корпус (конструктивное исполнение);

л) ориентированный граф. Возможны циклы, то есть «ребёнок» может быть «родителем» для того же элемента. Рёбра могут нести смысловую нагрузку, то есть сохраняется их значения. Рёбра между узлами имеют направление, порядок элементов важен. Например, применение в изделиях ЭКБ материала одного типа. Зл 99.9 – золотая проволока используется для соединения контактных площадок кристаллов с контактными площадками многовыводных рамок конкретных микросхем, также как в конкретных микросхемах используется соединение контактных площадок кристаллов с контактными площадками многовыводных рамок проволока Зл 99.9;

м) неориентированный граф. Подобен ориентированному графу, но рёбра между узлами не имеют строгого направления. Узлы можно читать и обходить в любом порядке. Например, математическая модель цепи соединения микросхемы, если эта модель отображает один из вариантов порядка соединения выводов элементов микросхемы.

Настраиваемые правила сопоставления, когда алгоритмы могут работать с различными форматами и структурами данных, примеры которых описаны выше, определяют гибкость алгоритма:

$$Flex = \frac{N}{\sum_{i=1}^{N} Var} * 100\%, \quad (7)$$

где N – число требований к гибкости, для алгоритмов сопоставления равно 11; Var – двоичное число, если требования выполняются, равно 1, иначе – 0. Если в (7) не выполняется ни одно требование, гибкость равна 0%. Алгоритм не может рассматриваться к применению.

5) Интеграция определяет совместимость с существующими рабочими процессами и программными средствами Системы. Для выполнения требований к интеграции должны выполняться требования:

а) обработка информации от нескольких источников. Интеграции требует подключение и понимание различных форматов, например, CSV или MySQL. Подключение к интерфейсу и проведение процесса сопоставления;

б) расширенная функция очистки данных. Интеграция предусматривает наличие инструмента профилирования данных на предмет несоответствий и ошибок. Пустые поля и поля со знаками препинания и символами не имеющие смысла, добавляют «шума» в данные и должны быть очищены;

в) очистка с помощью пользовательских регулярных выражений. Сложные строковые данные должны сопоставляться с помощью расширенных регулярных выражений, встроенных в данные. Или должны быть созданы пользовательские библиотеки выражений для дальнейшего использования;

г) стандартизация данных путем их разделения: при наличии нескольких наборов данных, необходима проверка и устранение несоответствиями в стандартах;

д) создание библиотеки слов в Системе: конкретные слова и аббревиатуры, которые заносятся в процессе сопоставления и предотвращение пометки Системой ненужных совпадений;

е) установка атрибутов и выполнение требований к: релевантности для выявления дубликатов или сходств; качеству данных для расстановки приоритетов атрибутов с помощью точных и согласованных данных; сспецифичность для атрибутов, которые предлагают четкие и надежные критерии соответствия;

ж) варьируемость для нечеткого совпадения; точного совпадения для идентичных значений или числового совпадение для параметров;

з) оценка совпадений, в результате принимается решение об объединении записей или сохранении нового набора записей в качестве основного.

Перечисленные требования к интеграции при их реализации в совокупности формируют показатель:

$$Integ = \frac{W}{\sum_{i=1}^{W} S} * 100\%, \quad (8)$$

где W – число требований к интеграции, для алгоритмов сопоставления W=8; S – двоичное число, если требования выполняются, S=1, иначе S=0. Если в (8) не выполняется ни одно требование, показатель интеграции равен 0%, и проверяемый алгоритм не может рассматриваться к применению.

ЗАКЛЮЧЕНИЕ

Представленные в статье методы расчета параметров прошли апробацию применительно к разработанному алгоритму сопоставления для нормализации данных системы формирования оптимальных предложений на выбор изделий ЭКБ в процессах разработки РЭА, доказавшие свою результативность в реально работающей системе «Дейтрон» от АО «ЦКБ «Дейтон», которая используется предприятиями радиоэлектроники.

Рассмотренный алгоритм адаптируется к изменяющимся данным, обрабатывает неструктурированный текст и сокращает ручной труд.

В алгоритме применены методы сопоставления данных и интеллектуального анализа данных. В то время как сопоставление данных фокусируется на сопоставлении элементов данных, интеллектуальный анализ данных фокусируется на выявлении скрытых закономерностей или знаний из данных, выполняя различные, но взаимодополняющие цели.

Сопоставление данных имеет важное значение для преобразования собранной информации в полезную, позволяя принимать обоснованные решения. Несмотря на существующие проблемы, развивающийся ландшафт инструментов и технологий сопоставления данных предлагает решения для преодоления этих препятствий.

Консолидированная и точная информация об ЭКБ способствует получению более глубокой аналитики, выявлению закономерностей, получая доступ к полному и согласованному представлению информации, принятию обоснованных решений на применение ЭКБ в РЭА на основе собранных данных.

Список литературы

1. *Pohan Lin*. Why is Data Normalization Important? // IEEE Computer Society, 2024.
2. *Anna Maricheva, Alexander Gedranovich*. What Is Data Normalization: A Perspective On Database Efficiency // Machine Learning Department at SoftTeco. 2025.
3. *Jawaria Irfan*. Mastering Data Normalization: A Comprehensive Guide. Learn Data Science, 2025 https://datasciencedojo.com/blog/data-normalization-importance/
4. *Peter Christen*. Data Matching. Concepts and Techniques for Record Linkage, Entity Resolution and Duplicate Detection. MA Group AG. Springer. 2012
5. *Forrest Brown*. Data Matching Software: Use Cases and Techniques. Profisee. London, UK, 2024
6. *Гагарина Л.Г., Рубцов Ю.В.* Особенности разработки метода классификации плоских QFN-корпусов для применения в составе автоматизированных систем техническойподготовки производства изделий микроэлектроники // Известия высших учебных заведений. Электроника. 2022. 27(3).

Рубцов Юрий Васильевич *– генеральный директор, АО «Центральное конструкторское бюро «Дейтон».*
E-mail: Rubtsov@Deyton.ru