Классификация подразумевает вычисление близости текста с другими текстами (представляющими классы) [1]. Текст, который наиболее близок с остальными текстами (классами) по содержанию, будет наиболее полно отражать смысл всех найденных статей [2].

Метод частотно-контекстного анализа текстовой информации отличается от частотного анализа тем, что он учитывает слова, используемые в тексте рядом с ключевыми.

Пример. Если информационный поток некоторого текста можно записать в виде F = (i3, i6, i7, i1, i2, i11, i9, i4, i10, i3, i5, i6, i7, i1, i8, i9, i4, i10, i5), то его структуру можно представить в виде графа (рис. 1):

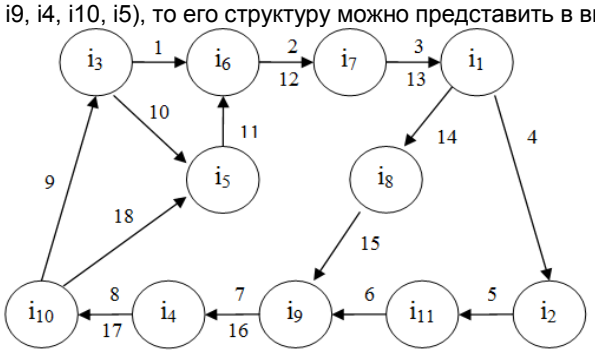

Рис. 1 – Структура, формируемая информационным потоком

Общая последовательность метода частотно-контекстного анализа выглядит следующим образом:

1) Моделирование текста и формирование его информационной структуры.

2) Выделение множества всех информационных элементов, ранжированных по их числу повторений в тексте.

3) Выделение множества ключевых элементов Sp.

4) Формирование уточняющего множества Ss на основе контекстного анализа информационных элементов множества Sp.

Предположим, что необходимо оценить эффективность метода частотно-контекстной классификации текстовой информации для решения задачи выбора текста, наиболее полно отражающего смысл некоторой темы (этот текст должен быть наиболее близок по содержанию к остальным текстам по теме).

Пусть есть множество текстов A_1, состоящее из текстов X_1, X_2, …, X_n. Пусть некоторым образом известно (например, согласно решению эксперта в предметной области), что текст, наиболее полно отражающий общую тему, – это текст Y. Некоторый метод вычисления близости текстов показывает, что суммарная близость с остальными текстами максимальна у текста X_{max}. Если $Y=X_{max}$, то текст выбран верно, иначе – неверно.

Проведя K подобных испытаний выбора текста, получим количество текстов R, классифицированных верно. Значение метрики, оценивающей эффективность применения метода, можно рассчитать по формуле:

$$V = R/K$$

Метрика принимает значения от 0 до 1.

Итак, была разработана метрика для оценки эффективности методов выбора текста, наиболее полно отражающего некоторую тему.

Список использованных источников:

1. Потараев В. В. Применение частотно-контекстной классификации текстовой информации при выборе текстов для изучения // Дистанционное обучение – образовательная среда XXI века: Материалы VIII международной научно-методической конференции – Минск, 2013 – с.340-341.
2. Тарасов, С.Д. Метод тематического связанного ранжирования для автоматического сводного реферирования новостных сообщений в задачах поддержки принятия управленческих решений/ С.Д. Тарасов // Вестник ВГУ.– 2010. №1. – С. 166–173.

ОБЕСПЕЧЕНИЕ АТРИБУТОВ ВЫСОКОЙ ДОСТУПНОСТИ И БЫСТРОГО ВОССТАНОВЛЕНИЯ ПОСЛЕ СБОЯ СИЛЬНО РАСПРЕДЕЛЕННЫХ СИСТЕМ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Базаревский В.Э., Базаревский Вл.Э.

Бранцевич П.Ю. − к. т. н., доцент

Рассматриваются архитектурные тактики, применимые для обеспечения следующих атрибутов качества сильно распределенных программных систем: высокая доступность, отказоустойчивость, быстрое восстановление

после сбоя, обеспечивающие незначительное снижение атрибутов масштабируемости и производительности.

Современная разработка программных систем зачастую сводится к решению одной из следующих задач: ручной или машинный сбор данных, очистка и преобразование данных, анализ данных. Одной из задач, которую необходимо решать в рамках всех указанных задач является проверка качества и целостности данных. Так, как ручной, так и машинный ввод данных может вносить ошибочные данные. Процессы очистки и преобразования данных так же могут вносить значительные ошибки, которые, однако зачастую могут быть обнаружены только на этапе финального анализа данных.

Следует заметить, что проверка качества данных в зависимости от методов может занимать значительное время, при этом, длительность проверки может сильно варьироваться от нескольких секунд, до нескольких часов (при этом, в зависимости от предметной области, не всегда можно дать априорную оценку длительности такого анализа).

Исходя из указанных выше требований, одним из способов решения задач такого долговременного и сложно прогнозируемого анализа может выступать разработка распределенных асинхронных систем на основе архитектурной модели сервисной шины (в английской литературе EnterpriseServiceBus, ESB). Такая архитектура позволяет обеспечить необходимый уровень производительности системы вцелом за счет легкой горизонтальной масштабируемости (в терминах облачных вычислений - эластичности) системы, когда по запросу в систему могут легко быть добавлены новые вычислительные мощности. Так, вычислительный кластер может быть резко увеличен с 3-4 вычислительных узлов, до 30-40 в случае пиковых нагрузок.

Однако такой подход в значительной степени усложняет процедуры поддержки системы вцелом, при этом значительно повышая вероятность сбоя того или иного вычислительного узла. Поэтому задачи мониторинга, централизованного отслеживания состояния всех вычислительных узлов системы, возможности централизованного обновления конфигураций всех элементов системы становятся одними из определяющих при разработке таких систем. Под централизованностью функционала управления системой при этом не подразумевается наличие единых точек отказа системы (singlepointoffailure), наличие которых резко бы понизило атрибуты отказоустойчивости и высокой доступности системы.

В качестве одной из наиболее интересных задач, подлежащих решению при разработке таких систем можно выделить задачу централизованного обновления конфигураций системы «на лету». Необходимость отсутствия единой точки отказа ограничивает возможность добавления единого узла конфигурации, которых бы хранил все системные настройки. При этом, все другие элементы системы: файловая система, база данных, очередь не могут выступать в качестве хранилища указанных настроек конфигурации, так как параметры доступа к ним сами являются настройками конфигурации. Оптимальным способом организации таких конфигураций нам видится реализация подхода master-slave с автоматическим выбором нового master-а из доступных slave при отказе предыдущего master-а.

Таким образом, задачи мониторинга состояния сильно распределенных систем являются ключевым функционалом, в значительной мере упрощающим процедуры поддержки работоспособности системы. При этом, добавление указанного функционала не должно в свою очередь снижать производительность, масштабируемость, отказоустойчивость и доступность системы.

Список использованных источников:

1. Басс Л. Архитектура программного обеспечения на практике. 2-е издание. / Басс Л., Клементс П., Кацман Р. СПб.: Питер, 2006. – 575 с.: ил.

АЛГОРИТМ СОЗДАНИЯ АВТОМАТИЧЕСКОГО АГЕНТА ДЛЯ ПОДДЕРЖКИ ПОЛЬЗОВАТЕЛЕЙ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Терещук А.В.

Серебряная Л.В. − к. т. н., доцент

В современном мире интернет оказывает все большее влияние на жизнедеятельность человека. Создается множество сложных программных продуктов, которые доступны клиентам по глобальной сети. Порог вхождения в такие системы довольно высок, поэтому для его снижения используются разные подходы. Одним из решений является поддержка пользователей в режиме онлайн чата. Для снижения затрат и увеличения прибыли поддержку пользователей можно организовать с использованием автоматического агента. В данной работе описан алгоритм реализации такого агента.

Целью применения автоматизации поддержки пользователей является уменьшение количества рутинных и однообразных действий, которые выполняют работники службы поддержки. Это достигается за счет того, что для решения большого количества типичных проблем создаются определенные ответы, которые передаются пользователям в соответствии с их запросами. Для соотнесения запросов и ответов подойдет алгоритм латентно-семантического анализа.

Латентно-семантический анализ отображает документы и отдельные слова в так называемое